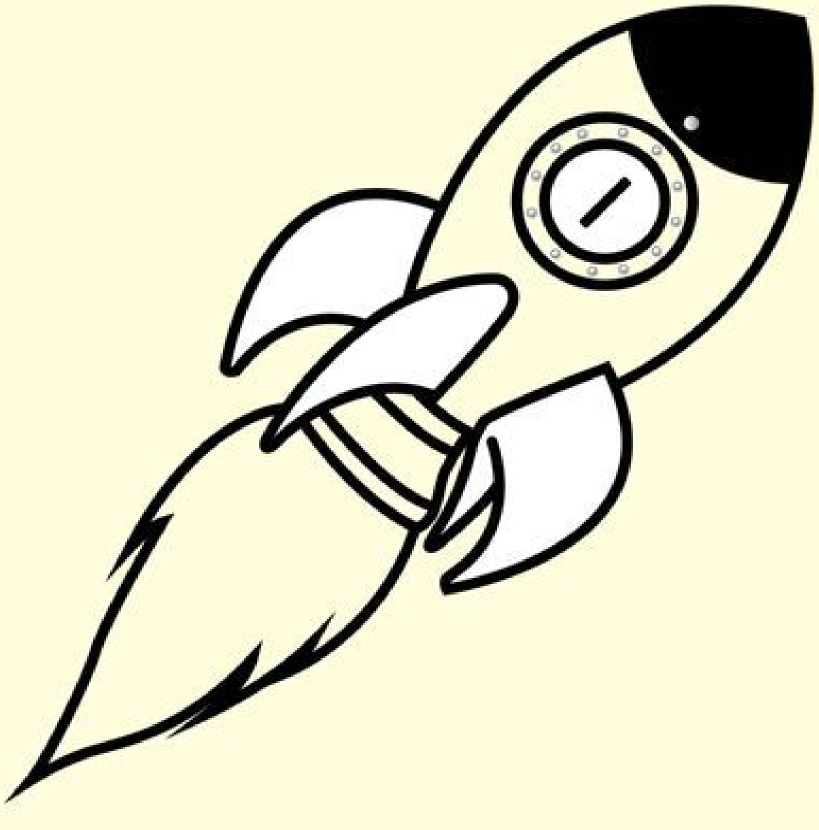
云栖社区 系列丛书

HZ BOOKS
华章IT

RocketMQ
实战与原理解析

杨开元◎著



APACHE ROCKETMQ

PRINCIPLE AND PRACTICE

阿里巴巴数据专家/RocketMQ源码贡献者撰写，RocketMQ官方开发团队鼎力推荐

机械工业出版社
China Machine Press

RocketMQ[illegible]

[illegible]

ISBN[illegible]978-7-111-60025-1

[illegible]2018[illegible]

[illegible]

[illegible]+ 86-10-68995265

[illegible]service@bbbvip.com

[illegible]www.hzmedia.com.cn

[illegible] @[illegible]

[illegible] [illegible]hzebook[illegible]

□□

[illegible]

[illegible]

[illegible]RocketMQ[illegible]IM[illegible]IoT[illegible]11[illegible]RocketMQ[illegible]3000[illegible]

[illegible]RocketMQ[illegible]2016[illegible]11[illegible]RocketMQ[illegible]Apache[illegible]RocketMQ[illegible]2017[illegible]9[illegible]Apache[illegible]RocketMQ[illegible]

[illegible]RocketMQ[illegible]Apache RocketMQ[illegible]RocketMQ[illegible]RocketMQ[illegible]——[illegible]

[illegible]Apache RocketMQ[illegible]RocketMQ[illegible]

——[illegible]Apache RocketMQ[illegible]

[illegible]

[illegible]

[illegible]Kafka[illegible]RocketMQ[illegible]RocketMQ[illegible]RocketMQ[illegible]

[illegible]RocketMQ[illegible]Java[illegible]

[illegible]RocketMQ[illegible]RocketMQ[illegible]2017[illegible]11[illegible]RocketMQ[illegible]TPS[illegible]5600[illegible]RocketMQ[illegible]“[illegible]”[illegible]

[illegible]

[illegible]RocketMQ[illegible]RocketMQ[illegible]RocketMQ[illegible]

[illegible]

·[illegible]

·[illegible]

·[illegible]

[illegible]

[illegible]RocketMQ[illegible]RocketMQ[illegible]

[illegible]

[illegible]

[illegible]RocketMQ[illegible]1[illegible]8[illegible]RocketMQ[illegible]

[illegible]RocketMQ[illegible]Consumer[illegible]Producer[illegible]RocketMQ[illegible]RocketMQ[illegible]

[illegible]9[illegible]13[illegible]RocketMQ[illegible]

[illegible]

[illegible]

[illegible]rocketmqqa@163.com[illegible]

[illegible]

[illegible]

[illegible]Leader[illegible]
RocketMQ[illegible]

[illegible]

[illegible]

[illegible]

[illegible]1[illegible] [illegible]

[illegible]RocketMQ[illegible]

1.1 [illegible]

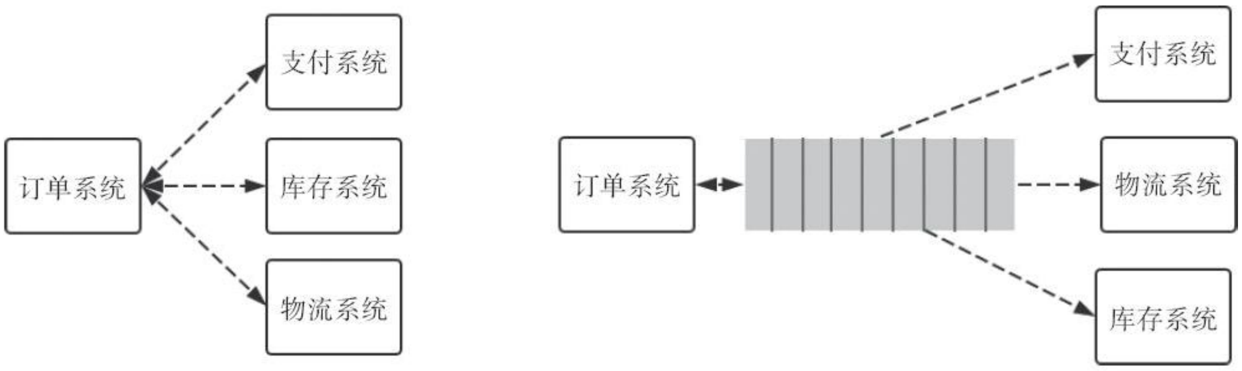
[illegible]“[illegible]”[illegible]“[illegible]”[illegible]IT[illegible]“[illegible]”[illegible]

1.1.1 [illegible]

[illegible]

[illegible]

[illegible]1-1[illegible]



[illegible]1-1 [illegible]

1.1.2 [illegible]

[illegible]0[illegible]

[illegible]

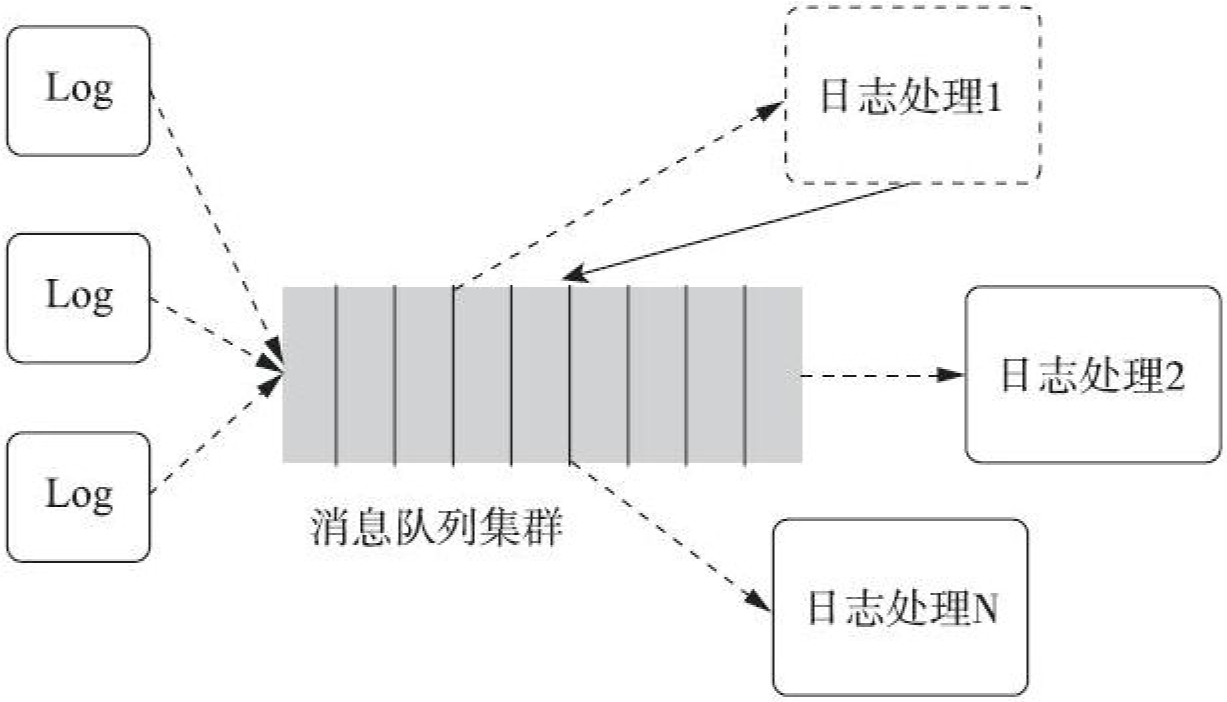
[illegible]QPS[illegible]QPS[illegible]QPS[illegible]QPS[illegible]

1.1.3 [illegible]

[illegible]

[illegible]1-2[illegible]Offset[illegible]

[illegible]



[illegible]1-2 [illegible]

1.2 RocketMQ[illegible]

[illegible]2007[illegible]Notify[illegible]2010[illegible]Napoli[illegible]2011[illegible]MetaQ[illegible]2012[illegible]RocketMQ[illegible]RocketMQ[illegible]Java[illegible]2016[illegible]Notify[illegible]MetaQ[illegible]RocketMQ[illegible]

[illegible]RocketMQ[illegible]Apache[illegible]RocketMQ[illegible]RocketMQ[illegible]2017[illegible]RocketMQ[illegible]TPS[illegible]5600[illegible]

[illegible]RocketMQ[illegible]Java[illegible]Kafka[illegible]Scala[illegible]RabbitMQ[illegible]Erlang[illegible]

1.3 [illegible]RocketMQ

[illegible]RocketMQ[illegible]
RocketMQ[illegible]

1.3.1 RocketMQ□□□□□□□□□

RocketMQ□Binary□□□□□□□□jar□□□□shell□□□□□□□□□□□□□□□□□http://rocketmq.apache.org/dowloading/releases/ □□□□□□□□□□□□□□

□□□□□64bit□Linux□Unix□Mac□Java□□□□□□JDK1.8□□□□□□GitHub□□□□□□□□□□□□□□□Maven 3.2.x□Git□

RocketMQ□□□□□□□□4.2.0□□□□Binary□□□□□□□□□□□□□

```
> unzip rocketmq-all-4.2.0-bin-release.zip -d ./rocketmq-all-4.2.0-binls
> cd rocketmq-all-4.2.0-bin/
```

□□□□□□□□□

```
LICENSE  NOTICE  README.md  benchmark/  bin/   conf/  lib/
```

LICENSE□NOTICE□README.md□□□□□□□□□□□□□□□□benchmark□□□□□benchmark□□□shell□□□bin□□□□□□□□□□RocketMQ□shell□□□Linux□□□□cmd□□□Windows□□□□□□□□□□□NameServer□□□mqnamesrv□□□Broker□□□mqbroker□□□□□□□mqadmin□□conf□□□□□□□□□□□□□□□□□□□□□broker□□□□□logback□□lib□□□□□□RocketMQ□□□□□□□□jar□□□□RocketMQ□□□□□jar□□□□Netty□commons-lang□FastJSON□□

1.3.2 [illegible]

[illegible]NameServer[illegible]Broker[illegible]

[illegible]NameServer[illegible]

```
> nohup sh bin/mqnamesrv &
> tail -f ~/Logs/rocketmqLogs/namesrv.Log
The Name Server boot success...
```

[illegible]Broker[illegible]

```
> nohup sh bin/mqbroker —n localhost:9876&
> tail -f ~/Logs/rocketmqLogs/broker.Log
The broker[%s, 192.168.0.233:10911] boot success...
```

1.3.3 [illegible]

[illegible]demo [illegible]demo[illegible]

[illegible]

```
 > export NAMESRV_ADDR=localhost:9876
 > sh bin/tools.sh org.apache.rocketmq.example.quickstart.Producer
SendResult [sendStatus=SEND_OK, msgId= ...

> sh bin/tools.sh org.apache.rocketmq.example.quickstart.Consumer
ConsumeMessageThread_%d Receive New Messages: [MessageExt...
```

1.3.4 [illegible]

[illegible]
NameServer[illegible]Broker[illegible]

[illegible]NameServer[illegible]Broker[illegible]

```
> sh bin/mqshutdown broker
The mqbroker(36695) is running...
Send shutdown request to mqbroker(36695) OK

> sh bin/mqshutdown namesrv
The mqnamesrv(36664) is running...
Send shutdown request to mqnamesrv(36664) OK
```

[illegible]RocketMQ[illegible]

1.4 □□□□

□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□RocketMQ□

[illegible]2[illegible] [illegible]

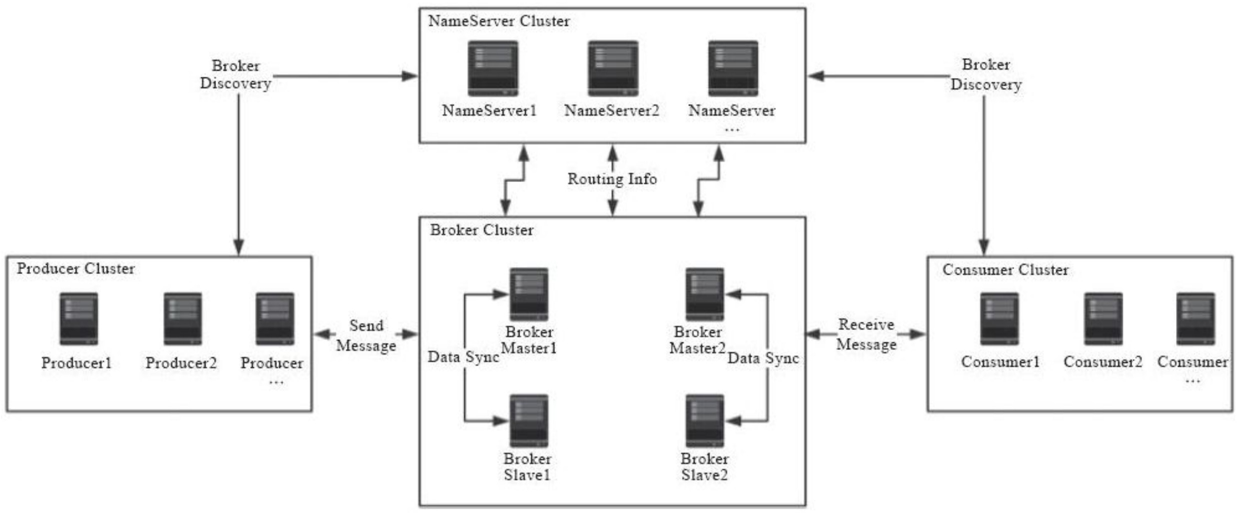
[illegible]RocketMQ[illegible]RocketMQ[illegible]RocketMQ[illegible]Consumer[illegible]Producer[illegible]

2.1 RocketMQ□□□□□□□

RocketMQ□□□RocketMQ□□□□□□□□□Producer□Consumer□Broker□NameServer□

□□RocketMQ□□□□□□□NameServer□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□□Producer□□□□□□□□□□□□Consumer□□□□□□□□□□□□□□□□□□□□□□□□□□Producer□Consumer□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NameServer□Broker□□□□Broker□□□□□□□Slave□



□2-1 RocketMQ□□□□□□□

□□□□□□□□□□□□□□□Topic□Message Queue□□Topic□□□□□□□□□□□□□□□□□□□□Topic□□□□□Topic□□□□□□□□□□Topic□□□□□□□□□□□□□□□□□Topic□□□Topic□□□□□□□□□□□□□Message Queue□

Message Queue□□□□□Partition□Topic□□□□Message Queue□□□
□□□□□□□□□Message Queue□□□□□□□□□□□□□□□Message Queue
□□□□□□□□

2.2 [illegible]

[illegible]RocketMQ[illegible]IP[illegible]192.168.100.131[illegible]192.168.100.132[illegible]

2.2.1 □□□□NameServer□Broker

□□□□□□□□□□□□NameServer□nohup sh bin/mqnamesrv &□□□□□□□□□□□□□□□NameServer□□□□□□□□“192.168.100.131□9876□192.168.100.132□9876”□

□□□□Broker□□□□□□□□□□□□□Master□□□Broker□□□Slave□□□Broker□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□□conf/2m-2s-sync□□□□□

1□192.168.100.131□□□Master Broker□□□□□

```
namesrvAddr=192.168.100.131:9876; 192.168.100.132:9876
brokerClusterName=DefaultCluster
brokerName=broker-a
brokerId=0
deleteWhen=04
fileReservedTime=48
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
listenPort=10911
storePathRootDir=/home/rocketmq/store-a
```

2□192.168.100.132□□□Master Broker□□□□□

```
namesrvAddr=192.168.100.131:9876; 192.168.100.132:9876
brokerClusterName=DefaultCluster
brokerName=broker-b
brokerId=0
deleteWhen=04
fileReservedTime=48
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
listenPort=10911
storePathRootDir=/home/rocketmq/store-b
```

3□192.168.100.131□□□Slave Broker□□□□□

```
namesrvAddr=192.168.100.131:9876; 192.168.100.132:9876
brokerClusterName=DefaultCluster
brokerName=broker-b
brokerId=1
deleteWhen=04
fileReservedTime=48
```

```
brokerRole=SLAVE
flushDiskType=ASYNC_FLUSH
listenPort=11011
storePathRootDir=/home/rocketmq/store-b
```

4□192.168.100.132□□□Slave Broker□□□□□

```
namesrvAddr=192.168.100.131:9876; 192.168.100.132:9876
brokerClusterName=DefaultCluster
brokerName=broker-a
brokerId=1
deleteWhen=04
fileReservedTime=48
brokerRole=SLAVE
flushDiskType=ASYNC_FLUSH
listenPort=11011
storePathRootDir=/home/rocketmq/store-a
```

□□□□□□□□□□□□□□Broker□

```
nohup sh ./bin/mqbroker -c  config_file &
```

□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□rocketmq-console□□□□192.168.100.131□□□RocketMQ-console□□□□□□□□□□□□192.168.100.131□8080□□□□□□□□□□□□□□□□□□

2.2.2 □□□□□

□□□□□□□Broker□□□□□□□□□□□□□

1□namesrvAddr=192.168.100.131□9876□
192.168.100.132□9876

NamerServer□□□□□□□□□□

2□brokerClusterName=DefaultCluster

Cluster□□□□□□□□□□□□□□□□□□□□□Cluster□□□Cluster□□□□□
□□□□

3□brokerName=broker-a

Broker□□□□Master□Slave□□□□□□□Broker□□□□□□□□□□□□□□
□Slave□□□Master□Slave□

4□brokerId=0

□□Master Borker□□□□□Slave□0□□Master□□□□0□□□□Slave□
ID□

5□fileReservedTime=48

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

6□deleteWhen=04

□fileReservedTime□□□□□□□□□□□□□□□□□□□□□04□□□□4□□

7□brokerRole=SYNC_MASTER

brokerRole□3□□SYNC_MASTER□ASYNC_MASTER□SLAVE□
□□□SYNC□ASYNC□□Master□Slave□□□□□□□□□□SYNC□□□□□

Slave□Master□□□□□□□□□□□□□□□□□

8□flushDiskType=ASYNC_FLUSH

flushDiskType□□□□□□□□SYNC_FLUSH□ASYNC_FLUSH□□□
□□□
□page_cache□□□□□□□□

9□listenPort=10911

Broker□□□□□□□□□□□□□□□□□Broker□□□□□□□□□□□□□□□

10□storePathRootDir=/home/rocketmq/store-a

□□□□□□□□□□□□□□□

□□□□□□□Broker□□□□□□□□□□□□□□□□□□Broker□□□□□□□
□□□□□□□□□□□Broker□□□□□□ip□□□□□□□□□□□
brokerIP1=47.98.41.234□□□□□□□□□□□Broker□□□□□□ip□□□

2.3 □□/□□□□□□

□□□□□□□□□□□□□□□□Java□□□□□RocketMQ Client□□□□□□□□□□2-1□□□□□□□□□□□□□□□□Sync□□□□□□□□

□□□□2-1 Producer□□□□

```
public class SyncProducer {
    public static void main(String[] args) throws Exception {
        //Instantiate with a Producer group name.
        DefaultMQProducer Producer = new
            DefaultMQProducer("please_rename_unique_group_name");
        producer.setNamesrvAddr("192.168.100.131:9876");
        //Launch the instance.
        Producer.start();
        for (int i = 0; i < 100; i++) {
            //Create a Message instance, specifying Topic, tag and Message body.
            Message msg = new Message("TopicTest" /* Topic */,
                "TagA" /* Tag */,
                ("Hello RocketMQ " +
                    i).getBytes(RemotingHelper.DEFAULT_CHARSET) /* Message body
*/
            );
            //Call send Message to deliver Message to one of brokers.
            SendResult sendResult = Producer.send(msg);
            System.out.printf("%s%n", sendResult);
        }
        //Shut down once the Producer instance is not longer in use.
        Producer.shutdown();
    }
}
```

□□□□□□□□□□DefaultMQProducer□□□□□□GroupName□NameServer□□□□□□□□□□□□□□□□□□Message□□□□□Producer□□□□□□□□□□□□□□□□□□□□□DefaultMQPush-Consumer□□□□□□□□□□□□□□□2-2□□□

□□□□2-2 Consumer□□□□

```
       /*
        * Instantiate with specified Consumer group name.
        */
        DefaultMQPushConsumer Consumer = new DefaultMQPushConsumer("please rename
to unique group name");
        /*
         * Specify name server addresses.
        Consumer.setNamesrvAddr("192.168.249.47:9876");
        /*
```

```
         * Specify where to start in case the specified Consumer group is a brand
new one.
         */
Consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
        //Consumer.setMessageModel(MessageModel.BROADCASTING);
        /*
         * Subscribe one more more Topics to consume.
         */
        Consumer.subscribe("TopicTest”, "*");
        /*
         *  Register callback to execute on arrival of Messages fetched from
brokers.
         */
        Consumer.registerMessageListener(new MessageListenerConcurrently() {
            public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
msgs,   ConsumeConcurrentlyContext context) {
                System.out.printf(Thread.currentThread().getName() + " Receive
New Messages: " + msgs + "%n");
                return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
            }
        });
        /*
         *  Launch the Consumer instance.
         */
        Consumer.start();
```

Consumer□Producer□□□□□GroupName□NameServer□□□□□□□□□□□□□□□□□Topic□□□□□□□□□□□□□□□

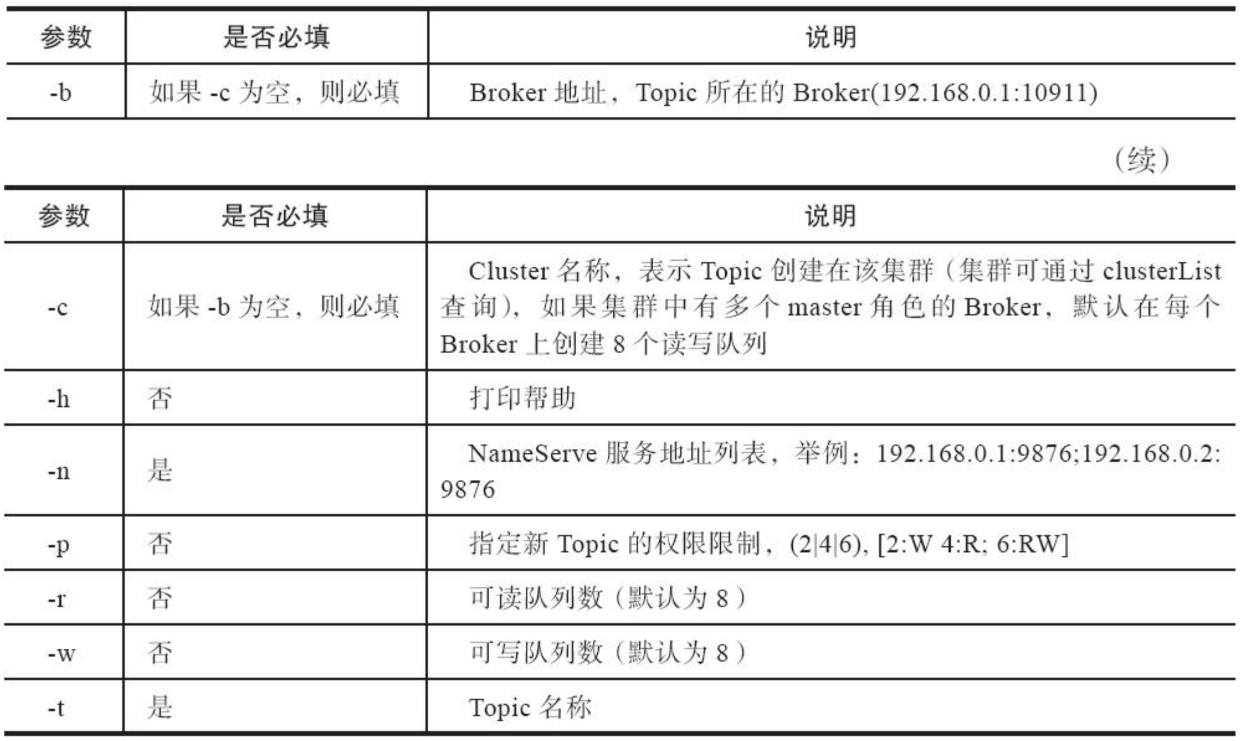
2.4 □□□□□□

MQAdmin□RocketMQ□□□□□□□□□□□□bin□□□□□□mqadmin□□□□□□□mqadmin□□□□□□□□□□□□Topic□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1.□□/□□Topic

□□□□□□□□□□□□□□Topic□□□□□□Topic□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□Topic□□□Topic□□□□updateTopic□□2-1□□□□□□□□□

□2-1 updateTopic



参数	是否必填	说明
-b	如果 -c 为空，则必填	Broker 地址，Topic 所在的 Broker(192.168.0.1:10911)
-c	如果 -b 为空，则必填	Cluster 名称，表示 Topic 创建在该集群（集群可通过 clusterList 查询），如果集群中有多个 master 角色的 Broker，默认在每个 Broker 上创建 8 个读写队列
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876
-p	否	指定新 Topic 的权限限制，(2|4|6), [2:W 4:R; 6:RW]
-r	否	可读队列数（默认为 8）
-w	否	可写队列数（默认为 8）
-t	是	Topic 名称

2.□□Topic

□□□/□□Topic□□□□□□Topic□□RocketMQ□□□□□□Topic□□□□□□□□deleteTopic□□2-2□□□□□□□□□

□2-2　deleteTopic

参数	是否必填	说明
-c	是	Cluster 名称，要删除的 Topic 所在的集群
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876; 192.168.0.2:9876
-t	是	Topic 名称

3.□□/□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Clustering□□□□□□□Topic□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Topic□□Topic□□□updateSubGroup□□2-3□□□□□□□□□

□2-3　updateSubGroup

参数	是否必填	说明
-b	如果 -c 为空，则必填	Broker 地址，创建订阅组所在的 Broker
-c	如果 -b 为空，则必填	Cluster 名称，创建订阅组所在的 Cluster
-d	否	是否容许广播方式消费
-g	是	订阅组名
-i	否	从哪个 Broker 开始消费
-m	否	是否容许从队列的最小位置开始消费（true|false），默认会设置为 true
-q	否	消费失败的消息放到一个重试队列，每个订阅组配置的重试队列数量
-r	否	重试消费最大次数，超过则投递到死信队列
-s	否	消费功能是否开启
-w	否	发现消息堆积后，将 Consumer 的消费请求重定向到另外一台 Broker 机器
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

4.□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□deleteSubGroup□□2-4□□□□□□□□□

□2-4　deleteSubGroup

参数	是否必填	说明
-b	如果 -c 为空，则必填	Broker 地址，删除订阅组所在的 Broker
-c	如果 -b 为空，则必填	Cluster 名称，删除订阅组所在的 Cluster
-g	是	订阅组名
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

5.□□Broker□□

Broker□□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□□□□Broker□□□□□□□□□□□□□□□updateBrokerConfig□□2-5□□□□□□□□□

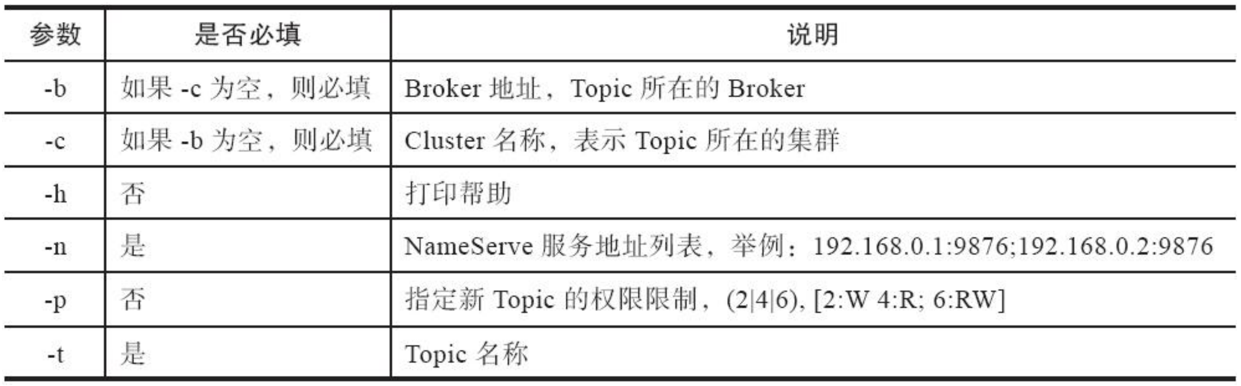
□2-5　updateBrokerConfig

参数	是否必填	说明
-b	如果 -c 为空，则必填	Broker 名称
-c	如果 -b 为空，则必填	Cluster 名称，该 Broker 所在的 Cluster
-k	是	Key 值
-v	否	Value 值
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

6.□□Topic□□□□□

RocketMQ□□□Topic□□□□□□□□□□□□□□Topic□□□□Topic□□□□□□□□□updateTopicPerm□□□□□□□2-6□□□□□□□□□

□2-6　updateTopicPerm

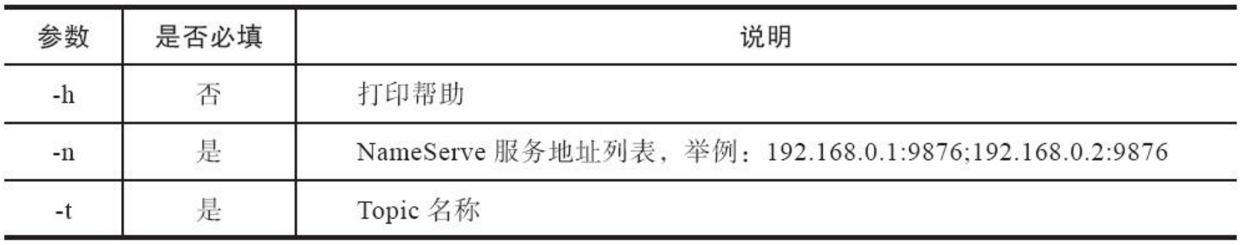


参数	是否必填	说明
-b	如果 -c 为空，则必填	Broker 地址，Topic 所在的 Broker
-c	如果 -b 为空，则必填	Cluster 名称，表示 Topic 所在的集群
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876
-p	否	指定新 Topic 的权限限制，(2|4|6)，[2:W 4:R; 6:RW]
-t	是	Topic 名称

7.□□Topic□□□□□

Topic□□□□□□□□□□Topic□□□Broker□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□TopicRoute□□2-7□□□□□□□□□

□2-7　TopicRoute

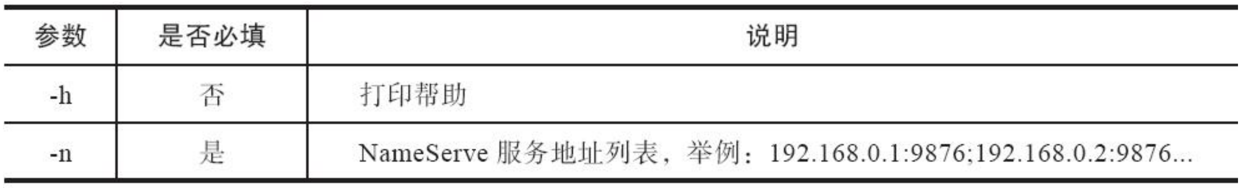


参数	是否必填	说明
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876
-t	是	Topic 名称

8.□□Topic□□□□

□□□□□TopicRoute□□□□□Topic□□□□□□□□□□□TopicList□□□□□□□□□Topic□□□□□2-8□□□□□□□□□

□2-8　TopicList

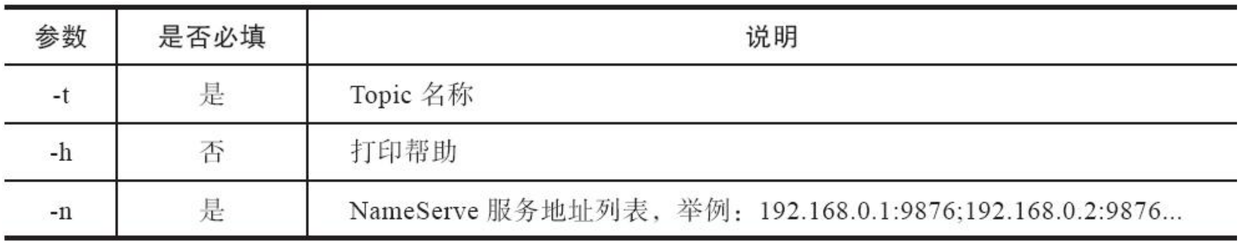


参数	是否必填	说明
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

9.□□Topic□□□□

[illegible]RocketMQ[illegible]Topic[illegible]TopicStats[illegible]2-9[illegible]

[illegible]2-9 TopicStats

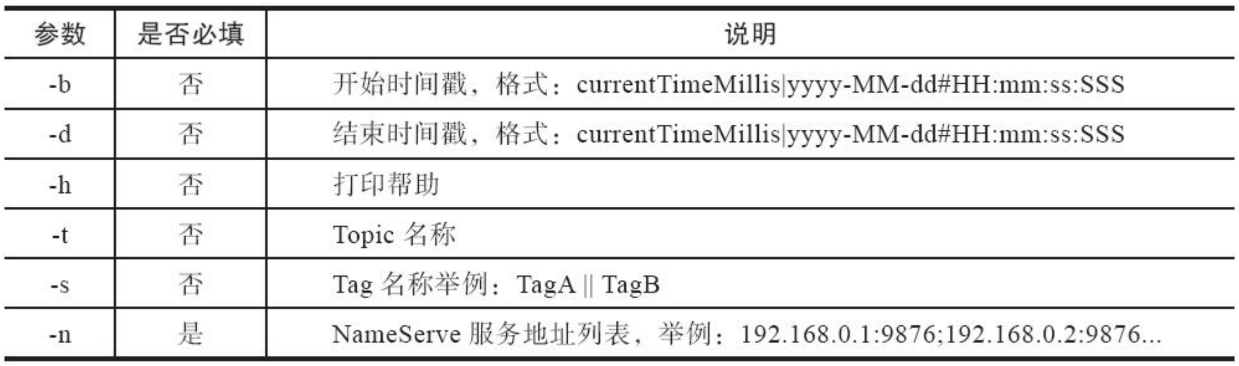


参数	是否必填	说明
-t	是	Topic 名称
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

10.[illegible]

[illegible]RocketMQ[illegible]printMsg[illegible]2-10[illegible]

[illegible]2-10 printMsg

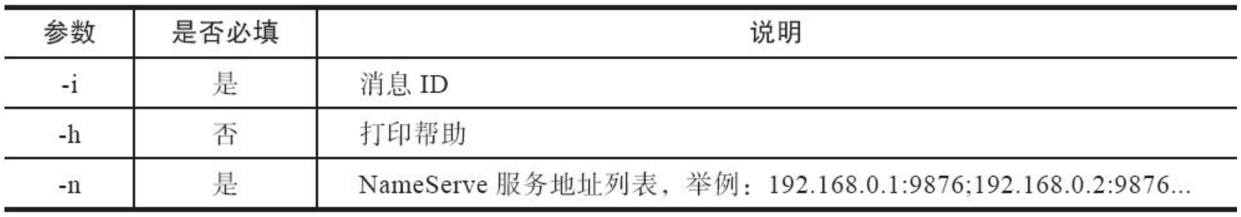


参数	是否必填	说明
-b	否	开始时间戳，格式：currentTimeMillis|yyyy-MM-dd#HH:mm:ss:SSS
-d	否	结束时间戳，格式：currentTimeMillis|yyyy-MM-dd#HH:mm:ss:SSS
-h	否	打印帮助
-t	否	Topic 名称
-s	否	Tag 名称举例：TagA || TagB
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

11.[illegible]ID[illegible]

[illegible]ID[illegible]ID[illegible]queryMsgById[illegible]2-11[illegible]

[illegible]2-11 queryMsgById

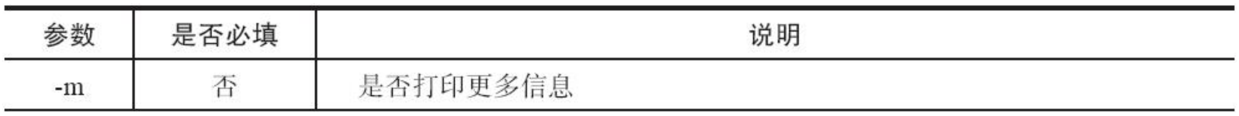


参数	是否必填	说明
-i	是	消息 ID
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

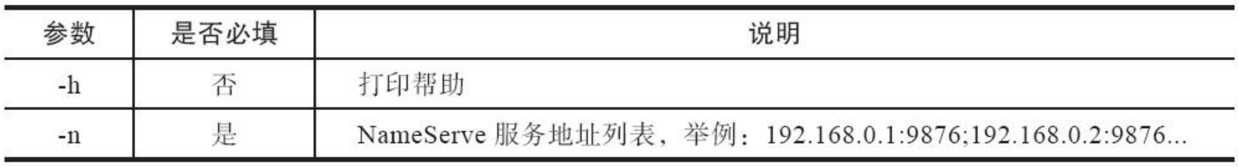
12.[illegible]

[illegible]clusterList[illegible]Broker[illegible]2-12[illegible]

[illegible]2-12　clusterList



（续）



参数	是否必填	说明
-m	否	是否打印更多信息
-h	否	打印帮助
-n	是	NameServe 服务地址列表，举例：192.168.0.1:9876;192.168.0.2:9876...

2.5 □□□□□□□□□□

□□RocketMQ□□□

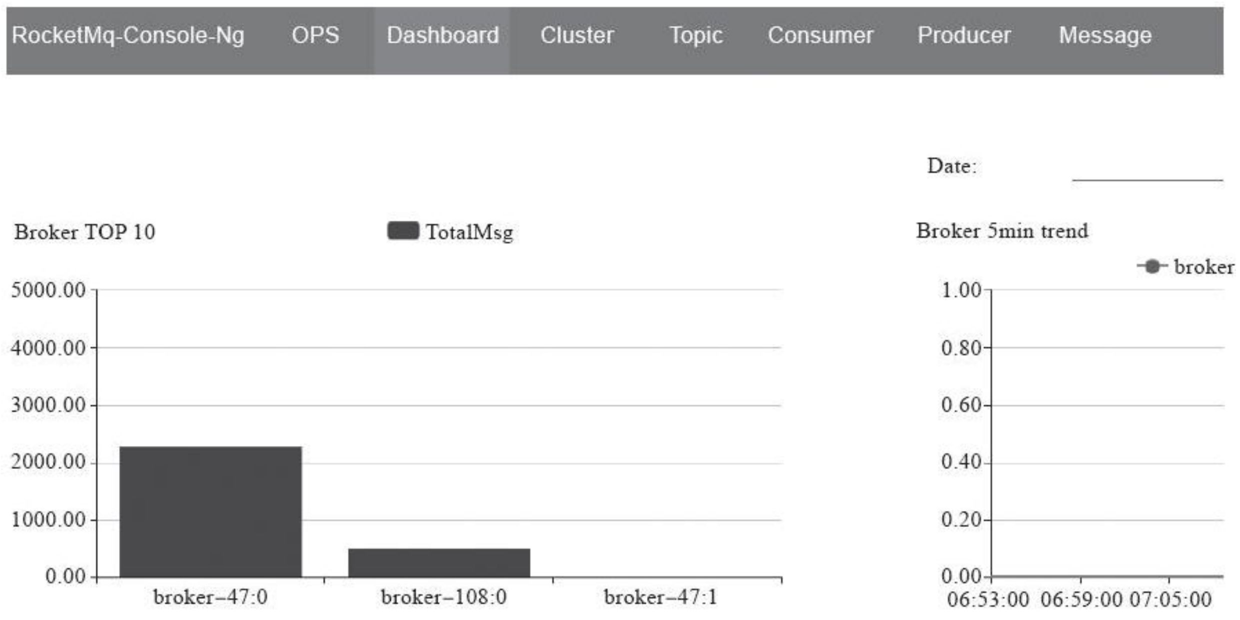
□□□□□□□□SpringBoot□□□□□□GitHub□□apache/rocketmq-externals□□□□□□https://github.com/apache/rocketmq-externals/tree/master/rocketmq-console □□

□□□□□□□□□□□□□□□□□□□□□

```
mvn spring-boot:run
```

□□□□□□jar□□□□java-jar□□□□

□□□□□□□□□□□□□server_ip_address□8080□server_ip_address□□□rocketmq-console□□□IP□□□□□□□□□□□□□



□2-2　rocketmq-console□□

2.6 [illegible]

[illegible]RocketMQ[illegible]QuickStart[illegible]
[illegible]
RocketMQ[illegible]
RocketMQ[illegible]

[illegible]3[illegible] [illegible]

[illegible]RocketMQ[illegible]Offset[illegible]Log[illegible]

3.1 [illegible]

[illegible]DefaultMQPushConsumer[illegible]DefaultMQPullConsumer[illegible]

3.1.1 DefaultMQPushConsumer□□□

□□DefaultMQPushConsumer□□Offset□□□□□□□DefaultMQPushConsumer□□□□□□□□□□□□□□org.apache.rocketmq.example.quickstart□□□□□□□□□□□□□□3-1□□□

□□□□3-1 DefaultMQPushConsumer□□

```
    public class QuickStart {
        public static void main(String[] args) throws InterruptedException,
    MQClientException {
            DefaultMQPushConsumer Consumer = new DefaultMQPushConsumer
    ("please_rename_unique_group_name_4");
    Consumer.setNamesrvAddr("name-server1-ip:9876;name-server2-ip:9876");
    Consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
            Consumer.setMessageModel(MessageModel.BROADCASTING);

            Consumer.subscribe("TopicTest", "*");
            Consumer.registerMessageListener(new MessageListenerConcurrently() {
                public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
    msgs,   ConsumeConcurrentlyContext context) {
                    System.out.printf(Thread.currentThread().getName() + " Receive
    New Messages: " + msgs + "%n");
                    return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
                }
            });
            Consumer.start();
        }
    }
```

DefaultMQPushConsumer□□□□□□□□□□□□□Consumer□GroupName□□□NameServer□□□□□□□□□□Topic□□□□□□□□□□□□□□□□

1□Consumer□GroupName□□□□□Consumer□□□□□□□□□□□□□□□GroupName□□□□□□□□MessageModel□□□□□□

RocketMQ□□□□□□□□□Clustering□Broadcasting□

·□Clustering□□□□□□□ConsumerGroup□GroupName□□□□□□□Consumer□□□□□□□□□□□□□□□□□□ConsumerGroup□□□□

Consumer□□□□□□□□□□□□□Topic□□□□□□□□□□□□□□□□□□

·□Broadcasting□□□□□□□ConsumerGroup□□□□Consumer□□□□□□□□Topic□□□□□□□□□□□□□□□□□□□□□□□Consumer□□□

2□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“ip1□port□ip2□port□ip3□port”□

3□Topic□□□□□□□□□□□□□□□□□□□□□□□□□□□Topic□□□□□□□□□□□□□□□□Tag□□□□□□□□□□Consumer.subscribe
□"TopicTest"□"tag1||tag2||tag3"□□□□□□Consumer□□
□“TopicTest”□□□tag1□tag2□tag3□□□Tag□□□□□□□□□□□□□□□□
Tag□□□□□□□null□□“*”□□□□□□□Topic□□□□□□

3.1.2 DefaultMQPushConsumer□□□□□

□□□□□□□□□□□DefaultMQPushConsumer□□□□□□

DefaultMQPushConsumer□□□□□□□DefaultMQPushConsumerImpl□□□□□□□□□□□□pullMessage□□□□□□PullCallBack□□□PullCallBack□□□□□switch□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3-2□□□

□□□□3-2 DefaultMQPushConsuer□□□□□

```
switch (pullResult.getPullStatus()) {
    case FOUND:
        ……
        break;
    case NO_NEW_MSG:
        ……
        break;
    case OFFSET_ILLEGAL:
        ……
        break;
    default:
        break;
}
```

DefaultMQPushConsuer□□□□□□□PullRequest□□□□□Default-MQPushConsumerImpl.this.executePullRequestImmediately□pullRequest□□□□□“PushConsumer”□□□“PullRequest”□□□□□□“□□□”□□□□Push□□□□□□□□□□□□□Pull□□□□□□□Push□□□□□□□

Push□□□Server□□□□□□□□□□□□□□□□Client□□□□□□□□□□□□□□□□□□Server□□□□Push□□□□□□□□□□□□□□□□□Server□□□□□□□□□□Server□□□□□□□Client□□□□□□□□□□Client□□□□□Server□□□□□Client□□□□□□Server□□□□□□□□□□□□□□□□□□

Pull□□□Client□□□□□Server□□□□□□□□□□Client□□□□□□□□□□□□□□□□□□□□□□□□□Pull□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□Pull□□□□□□□□Server□□□□□□□□□□□□□□□□□□□

“[illegible]”[illegible]Client[illegible]Server[illegible]Pull[illegible]3-3[illegible]3-4[illegible]

[illegible]3-3　[illegible]Pull[illegible]

```
PullMessageRequestHeader requestHeader = new PullMessageRequestHeader();
requestHeader.setConsumerGroup(this.ConsumerGroup);
requestHeader.setTopic(mq.getTopic());
requestHeader.setQueueId(mq.getQueueId());
requestHeader.setQueueOffset(Offset);
requestHeader.setMaxMsgNums(maxNums);
requestHeader.setSysFlag(sysFlagInner);
requestHeader.setCommitOffset(commitOffset);
requestHeader.setSuspendTimeoutMillis(brokerSuspendMaxTimeMillis);
requestHeader.setSubscription(subExpression);
requestHeader.setSubVersion(subVersion);
requestHeader.setExpressionType(expressionType);

------
PullResult pullResult = this.mQClientFactory.getMQClientAPIImpl().pullMessage(
    brokerAddr,requestHeader,timeoutMillis,communicationMode,pullCallback);
```

[illegible]requestHeader.setSuspendTimeoutMillis[illegible]brokerSus-pendMaxTimeMillis[illegible]Broker[illegible]15[illegible]Broker[illegible]

[illegible]3-4　“[illegible]”[illegible]

```
package org.apache.rocketmq.broker.longpolling
------
if (this.brokerController.getBrokerConfig().isLongPollingEnable()) {
    this.waitForRunning(5 * 1000);
} else {

this.waitForRunning(this.brokerController.getBrokerConfig().getShortPollingTimeMi
lls());
}
long beginLockTimestamp = this.systemClock.now();
this.checkHoldRequest();
long costTime = this.systemClock.now() - beginLockTimestamp;
if (costTime > 5 * 1000) {
    Log.info("[NOTIFYME] check hold request cost {} ms.", costTime);
}
```

[illegible]Broker[illegible]waitForRunning[illegible]5[illegible]Check[illegible]Broker[illegible]Check[illegible]Request[illegible]Broker-SuspendMaxTimeMillis[illegible]Broker[illegible]

[illegible]notifyMessageArriving[illegible]“[illegible]”[illegible]Broker[illegible]HOLD[illegible]Consumer[illegible]“[illegible]”[illegible]Consumer[illegible]Broker[illegible]Consumer[illegible]

[illegible]HOLD[illegible]Consumer[illegible]

3.1.3 DefaultMQPushConsumer□□□□□

□□□□PushConsumer□□□□□□□□PushConsumer□□□□□Pull□□

PushConsumer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3-5□□□

□□□□3-5 DefaultMQPushConsumer□□□□□□

```
this.consumeExecutor = new ThreadPoolExecutor(
    this.defaultMQPushConsumer.getConsumeThreadMin(),
    this.defaultMQPushConsumer.getConsumeThreadMax(),
    1000 * 60,
    TimeUnit.MILLISECONDS,
    this.consumeRequestQueue,
    new ThreadFactoryImpl("ConsumeMessageThread_"));
```

Pull□□□RocketMQ□□□□□□□□ProcessQueue□□□□□□□□□PushConsumer□□□□□□□□Message Queue□□□□□□□ProcessQueue□□□□□□□□Message Queue□□□□□□□□□□

ProcessQueue□□□□□□□□□□□TreeMap□□□□□□□TreeMap□□Message Queue□Offset□□Key□□□□□□□□□□Value□□□□□□□MessageQueue□□□□□□□□□□□□□□□□□□□□□□□□□□TreeMap□□□□□□□□

□□ProcessQueue□□□□□□□□□□□□□□□□□□□□□□Pull□□□□□□□□□□□□□□□□□□□□□□3-6□□□

□□□□3-6 PushConsumer□□□□□□□

```
long cachedMessageCount = processQueue.getMsgCount().get();
long cachedMessageSizeInMiB = processQueue.getMsgSize().get() / (1024 * 1024);
```

```
if (cachedMessageCount > this.defaultMQPushConsumer.getPullThresholdForQueue()) {
    this.executePullRequestLater(pullRequest,
PULL_TIME_DELAY_MILLS_WHEN_FLOW_CONTROL);
    if ((queueFlowControlTimes++ % 1000) == 0) {
        log.warn(
            "the cached message count exceeds the threshold {}, so do flow
control, minOffset={}, maxOffset={}, count={}, size={} MiB, pullRequest={},
flowControlTimes={}",
            this.defaultMQPushConsumer.getPullThresholdForQueue(),
processQueue.getMsgTreeMap().firstKey(), processQueue.getMsgTreeMap().lastKey(),
cachedMessageCount, cachedMessageSizeInMiB, pullRequest, queueFlowControlTimes);
    }
    return;
}
if (cachedMessageSizeInMiB > this.defaultMQPushConsumer.getPullThresholdSize-
ForQueue()) {
    this.executePullRequestLater(pullRequest,
PULL_TIME_DELAY_MILLS_WHEN_FLOW_CONTROL);
    if ((queueFlowControlTimes++ % 1000) == 0) {
        log.warn(
            "the cached message size exceeds the threshold {} MiB, so do flow
control, minOffset={}, maxOffset={}, count={}, size={} MiB, pullRequest={},
flowControlTimes={}",
            this.defaultMQPushConsumer.getPullThresholdSizeForQueue(),
processQueue.getMsgTreeMap().firstKey(), processQueue.getMsgTreeMap().lastKey(),
cachedMessageCount, cachedMessageSizeInMiB, pullRequest, queueFlowControl-Times);
    }
    return;
}
if (!this.consumeOrderly) {
    if (processQueue.getMaxSpan() >
this.defaultMQPushConsumer.getConsumeConcurrentlyMaxSpan()) {
        this.executePullRequestLater(pullRequest,
PULL_TIME_DELAY_MILLS_WHEN_FLOW_CONTROL);
        if ((queueMaxSpanFlowControlTimes++ % 1000) == 0) {
            log.warn(
                "the queue's messages, span too long, so do flow control,
minOffset={}, maxOffset={}, maxSpan={}, pullRequest={}, flowControlTimes={}",
                processQueue.getMsgTreeMap().firstKey(),
processQueue.getMsgTreeMap().lastKey(), processQueue.getMaxSpan(),
                pullRequest, queueMaxSpanFlowControlTimes);
        }
        return;
    }
}
```

[illegible]PushConsumer[illegible]
Offset[illegible]
ProcessQueue[illegible]

3.1.4 DefaultMQPullConsumer

□□DefaultMQPullConsumer□□□DefaultMQPushConsumer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□org.apache.rocketmq.example.simple□□□□□□□□□□□□□□□□3-7□□□

□□□□3-7 PullConsumer□□

```
public class PullConsumer {
    private static final Map<MessageQueue, Long> OFFSE_TABLE = new
HashMap<MessageQueue, Long>();

    public static void main(String[] args) throws MQClientException {
        DefaultMQPullConsumer Consumer = new DefaultMQPullConsumer
("please_rename_unique_group_name_5");
        Consumer.start();
        Set<MessageQueue> mqs =
Consumer.fetchSubscribeMessageQueues("TopicTest1");
        for (MessageQueue mq : mqs) {
            long Offset = Consumer.fetchConsumeOffset(mq, true);
            System.out.printf("Consume from the Queue: " + mq + "%n");
            SINGLE_MQ:
            while (true) {
                try {
                    PullResult pullResult =
                        Consumer.pullBlockIfNotFound(mq, null, getMessage-
QueueOffset(mq), 32);
                    System.out.printf("%s%n", pullResult);
                    putMessageQueueOffset(mq, pullResult.getNextBegin-Offset());
                    switch (pullResult.getPullStatus()) {
                        case FOUND:
                            break;
                        case NO_MATCHED_MSG:
                            break;
                        case NO_NEW_MSG:
                            break SINGLE_MQ;
                        case OFFSET_ILLEGAL:
                            break;
                        default:
                            break;
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
        Consumer.shutdown();
    }
    private static long getMessageQueueOffset(MessageQueue mq) {
        Long Offset = OFFSE_TABLE.get(mq);
        if (Offset != null)
            return Offset;
        return 0;
    }
```

```
        private static void putMessageQueueOffset(MessageQueue mq, long Offset) {
            OFFSE_TABLE.put(mq, Offset);
        }
    }
```

□□□□□□□□□□□□□□□Topic□□□Message Queue□□□□□□□□□□□□□□□□□□□□□□□□

□1□□□Message Queue□□□

□□Topic□□□□Message Queue□□□□□Consumer□□□Topic□□□□□□□□□□□□□□Message Queue□□□□□□□□□□□□□□□□□□□Message Queue□□□□□□

□2□□□Offsetstore

□□□Message Queue□□□□□□□□□□□□Offset□□□long□□□□□□□□□□□□□□□Offset□□□□□□□□□□□□□□□□Offset□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□3□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□FOUND□NO_MATCHED_MSG□NO_NEW_MSG□OFFSET_ILLEGAL□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□FOUNT□NO_NEW_MSG□□□□□□□□□□□□□□□□□□

□□□□□□□□while□true□□□□□□□□□□□□□□□□□□PullConsumer□□□□□□□□□□Message Queue□□□Offset□□□PullConsumer□□□□□□□□□□□□

3.1.5 Consumer□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□Consumer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Consumer□□Push□Pull□□□□□□□PullConsumer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Offset□□□□□□□□□□□□□□□□□□Offset□□□□□□□□□□□□□□□Message Queue□Offset□□□□□□□□□□□□□□

DefaultMQPushConsumer□□□□□□□shutdown□□□□□□□□□□□□□□Offset□□□□□□□□□Consumer□□□□□□□□□□□

PushConsumer□□□□□□□□□□□□□□□□□□□□NameServer□□Topic□□□□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□WARN□□□□□□□□□□□□□□□□□□□□□DefaultMQPushConsumer□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□

□□□DefaultMQPushConsumer□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□NameServer□Broker□□□□□□□□□□□□□□□□□□□□DefaultMQPushConsumer□□□DefaultMQPushConsumer□□□□□□□□□□kill□Broker□NameServer□□□□□□□□□□□□DefaultMQPushConsumer□□□

□□□□□DefaultMQPushConsumer□□□□□□□□□□□□□□□□□□□□□□□□□□Consumer.start□□□□□□□□Consumer.fetchSubscribeMe-ssageQueues□"TopicName"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□MQClientException□□□

3.2 [illegible]

[illegible]

3.2.1　DefaultMQProducer

□□□□□□□□□□□□□DefaultMQProducer□□□□□□□□□□□□□□□□□□□□□□□□3-8□□□

□□□□3-8　DefaultMQProduce□□

```
public class ProducerQuickStart {
    public static void main(String[] args) throws MQClientException,
InterruptedException {
    DefaultMQProducer producer = new
DefaultMQProducer("please_rename_unique_group_name");
    producer.setInstanceName("instance1");
    producer.setRetryTimesWhenSendFailed(3);
producer.setNamesrvAddr("name-server1-ip:9876;name-server2-ip:9876");
        Producer.start();
        for (int i = 0; i < 1000; i++) {
            try {
                Message msg = new Message("TopicTest" /* Topic */,
                    "TagA" /* Tag */,
                    ("Hello RocketMQ " +
i).getBytes(RemotingHelper.DEFAULT_CHARSET) /* Message body */
                );
                Producer.send(msg, new SendCallback() {
                    public void onSuccess(SendResult sendResult) {
                        System.out.printf("%s%n", sendResult);
                        sendResult.getSendStatus();
                    }
                    public void onException(Throwable e) {
                        e.printStackTrace();
                    }
                });
            } catch (Exception e) {
                e.printStackTrace();
                Thread.sleep(1000);
            }
        }
        producer.shutdown();
    }
}
```

□□□□□□□□□□□□

1□□□Producer□GroupName□

2□□□InstanceName□□□□Jvm□□□□□□Producer□□□□□□□□□□□InstanceName□□□□□□□□□□□□□□□□□“DEFAULT”□

3[illegible]

4[illegible]NameServer[illegible]

5[illegible]

[illegible]2[illegible]FLUSH_DISK_TIMEOUT[illegible]FLUSH_SLAVE_TIMEOUT[illegible]SLAVE_NOT_AVAILABLE[illegible]SEND_OK[illegible]

·FLUSH_DISK_TIMEOUT[illegible]Broker[illegible]SYNC_FLUSH[illegible]

·FLUSH_SLAVE_TIMEOUT[illegible]Broker[illegible]SYNC_MASTER[illegible]

·SLAVE_NOT_AVAILABLE[illegible]FLUSH_SLAVE_TIMEOUT[illegible]Broker[illegible]SYNC_MASTER[illegible]Slave[illegible]Broker[illegible]

·SEND_OK[illegible]Slave[illegible]Slave[illegible]SEND_OK[illegible]

[illegible]

3.2.2 □□□□□□

RocketMQ□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□Message□□□□□□setDelayTimeLevel□int level□□
□1s/5s/10s/30s/1m/2m/3m/4m/5m/6m/7m/8m/9m/10m/20m/30m/1h/2h□□□□setDelayTimeLevel□3□□□□□10s□

3.2.3 □□□□□□□□

□□Topic□□□□Message Queue□□□□□Producer□□□□□□□□
Producer□□□□□□Message Queue□□□□□Consumer□□□□□□□□□□
□□□□□□□□□□□□□□□□Message Queue□□□□□□□□□□□□□□□□□□□□□
Message Queue□□□□Consumer□□□□□□□

□□□□□□□□□□□□□□□□□Message Queue□□□□□□□□□□□□□□□□□
□Message Queue□□□□□□□□□□Message-QueueSelector□□□□□□
3-9□□□

□□□□3-9   MessageQueueSelector□□

```
public class OrderMessageQueueSelector implements MessageQueueSelector {
     public MessageQueue select(List<MessageQueue> mqs, Message msg,
Object orderKey) {
    int id = Integer.parseInt(orderKey.toString());
    int idMainIndex  = id/100;
    int size = mqs.size();
    int index = idMainIndex%size;
    return mqs.get(index);
  }
}
```

□□□□□□□□□MessageQueueSelector□□□□□□□□□□public
SendResult send□Message msg□MessageQueueSelector
selector□Object arg□□□□□□□□□□□MessageQueueSelector□□□
□□□□□□□Object□□□□□□□Message□□□□□□□□□□□□□Message
Queue□□□□□□□Message Queue□

3.2.4 □□□□□

RocketMQ□□A□□□□□□□□□□□□B□□□□□□□A□□□□“B□□□□□□□□□”□□□□□□□“□A□□□□□□□□”□□□□□□□□□□□□□□□

RocketMQ□□□□□□□□□□□□□□□□□TransactionMQProducer□□□□□□□□□□□□□□“□□□B□□□□□□□□□”□□□□□□□□□□□A□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□B□□□□□□□□□”□□□□□commit□□rollback□□□□□□□□

1□□□□□RocketMQ□□“□□□”□□□

2□RocketMQ□□□□“□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

3□□□□□□□□□□□□□□□

4□□□□□□□□□□□□□□□RocketMQ□□□□□□□Commit□□Rollback□□□□RocketMQ□□Commit□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Rollback□□□□□□□□□□□□□□□□□□□□□□□□

5□□□□□□□□□□□□4□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□“□□□”□□□□□□□□□

6□□□□□□□□□□□□□□□□□□□□□□□□Producer□□□□□□□□□□□□□□□Producer□□□□Group□□□□Producer□□□□□□□□□□□□□□□□□□□□□Commit□Roolback□□□

7□RocketMQ□□□□□□□□□□□□4□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□□4□□□□□□□□□□□□□□□□□□□□□□□Catch□□□□□□□□□□□□□□□□RocketMQ□4.x□□□□□□□□□□□□□□□□□□□□□Class□□□□□□□□□□□□□□□□□□□□□□□□

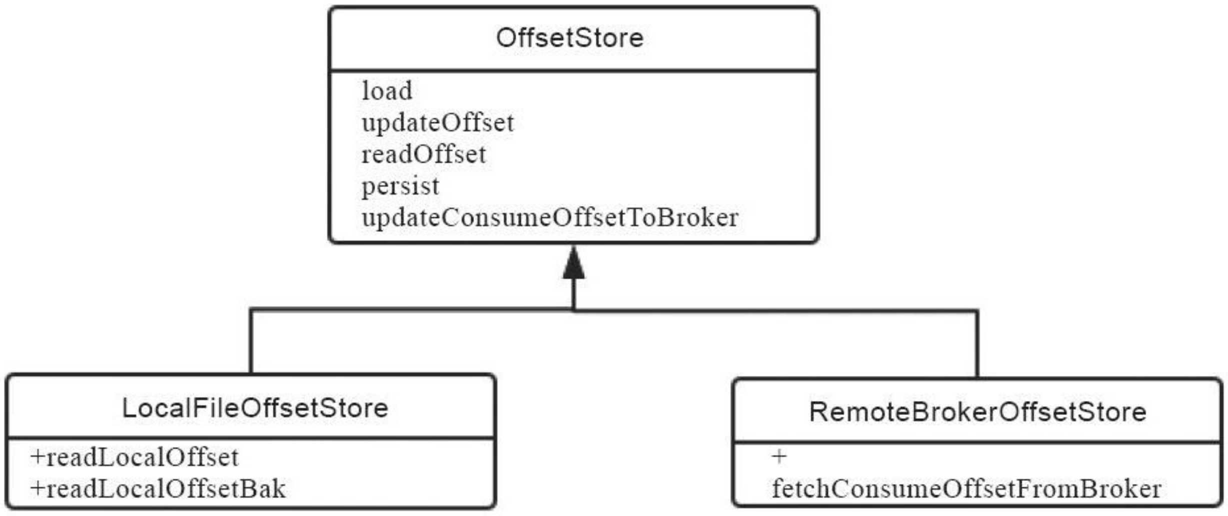
□□□□□□□□□□□□□□□□□□□□□□□LocalTransaction-Executer□□□□□□□□3□□□□□□□□□□□LocalTransactionState.ROLLBACK_MESSAGE□□LocalTransactionState.COMMIT_MESSAGE□□□□□□□□TransactionMQProducer□□□□□□DefaultMQProducer□□□□□□□□□□□Producer□□□□□□□□DefaultMQProducer□□□□□□□□□□□□□□□□□□□□□□□TransactionCheckListener□□□□□5□□MQ□□□□□□□□□□□LocalTransactionState.ROLLBACK_MESSAGE□□LocalTransactionState.COMMIT_MESSAGE□

3.3 [illegible]

[illegible]Offset[illegible]Offset[illegible]Offset[illegible]

[illegible]Offset[illegible]RocketMQ[illegible]Topic[illegible]Topic[illegible]Message Queue[illegible]Offset[illegible]Topic[illegible]Message Queue[illegible]Offset[illegible]Consumer[illegible]

[illegible]3-1[illegible]Offset[illegible]Broker[illegible]DefaultMQPushConsumer[illegible]CLUSTERING[illegible]Consumer group[illegible]Broker[illegible]Offset[illegible]RemoteBrokerOffsetStore[illegible]



[illegible]3-1 OffsetStore[illegible]

[illegible]DefaultMQPushConsumer[illegible]BROADCASTING[illegible]Consumer[illegible]Topic[illegible]Consumer[illegible]RocketMQ[illegible]LocalFileOffsetStore[illegible]Offset[illegible]

OffsetStore[illegible]Json[illegible]

[illegible]3-10 Offsetstore[illegible]

```
{"OffsetTable":{{"brokerName":"localhost", "QueueId":1,"Topic":"broker1" }: 1,{
"brokerName":"localhost", "QueueId":2,"Topic":"broker1" }:2, {
"brokerName":"localhost", "QueueId":0, "Topic":"broker1" }:3 } }
```

□□□DefaultMQPushConsumer□□□□□□□□□□OffsetStore□□□□□□□PullConsumer□□□□□□□□□OffsetStore□□□3.1.4□□PullConsumer□□□□□□□□Offset□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Offset□□□□□□□□□□□□□□□□□□□□□□□□Offset□□□□□□□□LocalFileOffsetStore□□□□□□□□□□□3-11□□□

□□□□3-11　□□□□□□□OffsetStore

```
public class LocalOffsetStoreExt {
    private final String groupName;
    private final String storePath;
    private ConcurrentMap<MessageQueue, AtomicLong> OffsetTable =
            new ConcurrentHashMap<MessageQueue, AtomicLong>();
    public LocalOffsetStoreExt(String storePath, String groupName) {
        this.groupName = groupName;
        this.storePath = storePath;
    }
    public void load() {
        OffsetSerializeWrapper OffsetSerializeWrapper = this.readLocal-Offset();
        if (OffsetSerializeWrapper != null &&
OffsetSerializeWrapper.getOffsetTable() != null) {
            OffsetTable.putAll(OffsetSerializeWrapper.getOffsetTable());
            for (MessageQueue mq :
OffsetSerializeWrapper.getOffsetTable().keySet()) {
                AtomicLong Offset = OffsetSerializeWrapper.getOffset-
Table().get(mq);
                System.out.printf("load Consumer's Offset, {} {} {} \n",
this.groupName, mq, Offset.get());
            }
        }
    }
    public void updateOffset(MessageQueue mq, long Offset) {
        if (mq != null) {
            AtomicLong OffsetOld = this.OffsetTable.get(mq);
            if (null == OffsetOld) {
                this.OffsetTable.putIfAbsent(mq, new AtomicLong(Offset));
            } else {
                OffsetOld.set(Offset);
            }
        }
    }
    public long readOffset(final MessageQueue mq) {
        if (mq != null) {
            AtomicLong Offset = this.OffsetTable.get(mq);
            if (Offset != null) {
                return Offset.get();
            }
        }
        return 0;
    }
    public void persistAll(Set<MessageQueue> mqs) {
        if (null == mqs || mqs.isEmpty())
```

```
            return;
        OffsetSerializeWrapper OffsetSerializeWrapper = new Offset-
SerializeWrapper();
        for (Map.Entry<MessageQueue, AtomicLong> entry : this.OffsetTable.
            entrySet()) {
            if (mqs.contains(entry.getKey())) {
                AtomicLong Offset = entry.getValue();
                OffsetSerializeWrapper.getOffsetTable().put(entry.getKey(),
Offset);
            }
        }
        String jsonString = OffsetSerializeWrapper.toJson(true);
        if (jsonString != null) {
            try {
                MixAll.string2File(jsonString, this.storePath);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    private OffsetSerializeWrapper readLocalOffset() {
        String content = null;
        try {
            content = MixAll.file2String(this.storePath);
        } catch (IOException e) {
            e.printStackTrace();
        }
        if (null == content || content.length() == 0) {
            return null;
        } else {
            OffsetSerializeWrapper OffsetSerializeWrapper = null;
            try {
                OffsetSerializeWrapper =
                        OffsetSerializeWrapper.fromJson(content, Offset-
SerializeWrapper.class);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return OffsetSerializeWrapper;
        }
    }
}
```

[illegible]OffsetStore[illegible]Consumer[illegible]DefaultMQPushConsumer[illegible]setConsumeFromWhere [illegible]ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET[illegible]Offset[illegible]CONSUME_FROM_FIRST_OFFSET[illegible]Consumer.setConsumeF-romWhere [illegible]ConsumeFromWhere.CONSUME_FROM_TIMESTAMP[illegible]Consumer.setConsumeTimestamp[illegible]"20131223171201"[illegible]

[illegible]Offset Store[illegible]DefaultMQPushConsumer[illegible]BROADCASTING[illegible]Broker[illegible]Topic[illegible]ConsumerGroup[illegible]Offset[illegible]Offset[illegible]ConsumeFromWhere[illegible]Consumer Group[illegible]Consumer[illegible]Offset[illegible]ConsumeFromWhere[illegible]

3.4 □□□□□□□

Log□□□□□□□□□□□□□□□□□RocketMQ□□□Log□□□□□□
${user.home}/Logs/rocketmqLogs□Log□□□□□□□□□□JVM□□□
□□□□□□□□□□□□□□□□□□□□□□

RocketMQ□□□□□□□org.apache.rocketmq.Client.Log
ClientLogger□□□□□□□□□□□□□□□□□□□□□□RocketMQ Client□
Log level□□□□-Drocketmq.Client.LogLevel□□□□□□□□□□□□□
System.setProperty□"rocketmq.Client.LogLevel"□"WARN"□□□
□□

RocketMQ□Log□□□□□slf4j□□□□Logback□Log4j□RocketMQ
Client□□□□Logback□□□□□□□□□□Logback□□□□□□Logback□□□
□□□□□□□□□□□□□

□□□□□maven□□□□□□□□□□□□□□□□Log□□□

□□□□□rocketmq.Client.Log.loadconfig□□□□□false□□□□□□□
□□System.setProperty
□"rocketmq.Client.Log.loadconfig"□"false"□□□□□□JVM□□□□□-
D□□□□□□□□Logback.xml□□maven□□□resources□□□□□
Logback.xml□□□□□□□□RocketMQ□□□□□□□□□STDOUT□□□□□
□□RocketMQ□□□□□□□□□console□□□□□□□□□□□□□□□3-12□□□

□□□□3-12　Logback.xml□□

```
<configuration>
    <appender name="RocketmqClientAppender"
              class="ch.qos.Logback.core.rolling.RollingFileAppender">
        <file>/Users/mark.yky/IdeaProjects/mqClientest/Logs/rocketmq_Client.
Log</file>
        <append>true</append>
        <rollingPolicy class="ch.qos.Logback.core.rolling.FixedWindow-
RollingPolicy">

<fileNamePattern>/Users/mark.yky/IdeaProjects/mqClientest/otherdays/rocketmq_Clie
nt.%i.Log
            </fileNamePattern>
            <minIndex>1</minIndex>
```

```
                <maxIndex>20</maxIndex>
            </rollingPolicy>
            <triggeringPolicy
                  class="ch.qos.Logback.core.rolling.SizeBasedTriggeringPolicy">
                <maxFileSize>100MB</maxFileSize>
            </triggeringPolicy>
            <encoder>
                <pattern>%d{yyy-MM-dd HH:mm:ss,GMT+8} %p %t - %m%n</pattern>
                <charset class="java.nio.charset.Charset">UTF-8</charset>
            </encoder>
        </appender>
        <appender name="STDOUT" class="ch.qos.Logback.core.ConsoleAppender">
            <layout class="ch.qos.Logback.classic.PatternLayout">
                <Pattern>
                    %d{yyy-MM-dd HH:mm:ss,GMT+8} %p %t - %m%n
                </Pattern>
            </layout>
        </appender>
        <Logger name="RocketmqCommon" additivity="false">
            <level value="DEBUG"/>
            <appender-ref ref="RocketmqClientAppender"/>
        </Logger>
        <Logger name="RocketmqRemoting" additivity="false">
            <level value="DEBUG"/>
            <appender-ref ref="RocketmqClientAppender"/>
        </Logger>
        <Logger name="RocketmqClient" additivity="false">
            <level value="DEBUG"/>
            <appender-ref ref="RocketmqClientAppender"/>
            <appender-ref ref="STDOUT"/>
        </Logger>
    </configuration>
```

□□□□□□Log□□□□□□□□□□□□□□□□□□□□□□Level□□□□□□□□□□□□ □□□□□□□□□□□□□□Logback□□□□□□□□ https://Logback.qos.ch/manual/configuration.html □

3.5 [illegible]

[illegible]Consumer[illegible]Producer[illegible]Consumer[illegible]Producer[illegible]Offset[illegible]Log[illegible]Offset[illegible]RocketMQ[illegible]Log[illegible]RocketMQ[illegible]NameServer[illegible]

□4□　□□□□□□□□□□□

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□IP□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□Producer□Consumer□□□□□□□□□□□□□
NameServer□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□
□□□□□□□NameServer□□□□□□

4.1 NameServer[illegible]

NameServer[illegible]NameServer[illegible]NameServer[illegible]

NamServer[illegible]NameServer[illegible]NameServer[illegible]NameServer[illegible]Broker[illegible]Topic[illegible]NameServer[illegible]

4.1.1 □□□□□□□□□

□org.apache.rocketmq.namesrv.routeinfo□RouteInfoManager□□□□□□□□□□□□□□□□□□□□□□□□□

·private final HashMap<String/*topic*/□List<QueueData>>topicQueueTable

topicQueueTable□□□□□Key□Topic□□□□□□□□□□Topic□□□□□□Value□□QueueData□□□□□□□□□□□□Topic□□□□□Master Broker□□□□QueueData□□□□Broker□□□□□□queue□□□□□□□□□□

·private final HashMap<String/*BrokerName*/□BrokerData>Broker-AddrTable

□BrokerName□□□□□□□□□Broker□□□□□□□□□□□Master□□□Slave□□□□□□□□□□BrokerName□□□□□□□□□□□□□Cluster□□□□□Master Broker□□□Slave Broker□□□□□□

·private final HashMap<String/*ClusterName*/□Set<String/*BrokerName*/>>ClusterAddrTable

□□□□□□□Cluster□□□□□□□□□□□□□□Cluster□□□□□□□BrokerName□□□□□□

·private final HashMap<String/*BrokerAddr*/□BrokerLiveInfo>Broker-LiveTable

□□□□□BrokerAddrTable□□□□□□□□□□□□□□□□□□Key□BrokerAddr□□□□□□□□□□□□BrokerAddrTable□□Key□BrokerName□□□□□□BrokerName□□□□□BrokerLiveTable□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□Broker□□□□□□□Broker□□□□□□

·private final HashMap<String/*BrokerAddr*/□
List<String>/*Filter Server*/>filterServerTable

Filter Server□□□□□□□□RocketMQ□□□□□□□□□□□□□Broker□□□□□□□□Filter Server□□□□□□Key□Broker□□□□Value□□□□Broker□□□□□Filter Server□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□

4.1.2 □□□□□□

□□□□□□□□NameServer□□□□□□Broker□□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□NameServer□□□□□□□NameServer□□□□□□DefaultRequest-Processor□□□org.apache.rocketmq.namesrv.routeinfo□BrokerHousekeepingService□□□□□□□□4-1□□□

□□□□4-1 Channel□□□□□□□□□

```
@Override
public void onChannelClose(String remoteAddr, Channel channel) {
    this.namesrvController.getRouteInfoManager().onChannelDestroy (remoteAddr,
channel);
}
@Override
public void onChannelException(String remoteAddr, Channel channel) {
    this.namesrvController.getRouteInfoManager().onChannelDestroy (remoteAddr,
channel);
}
@Override
public void onChannelIdle(String remoteAddr, Channel channel) {
    this.namesrvController.getRouteInfoManager().onChannelDestroy (remoteAddr,
channel);
}
```

□NameServer□Broker□□□□□□□□□onChannelDestroy□□□□□□□□□□Broker□□□□□□□□

NameServer□□□□□□□□□□□□□Broker□NameServer□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□4-2□□□

□□□□4-2 □□Check Broker□□□

```
this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
    @Override
    public void run() {
        NamesrvController.this.routeInfoManager.scanNotActiveBroker();
    }
}, 5, 10, TimeUnit.SECONDS);
```

[illegible]10[illegible]2[illegible]Broker[illegible]

4.2 [illegible]

[illegible]Topic[illegible]NameServer[illegible]
NameServer[illegible]

4.2.1 □□□□□□□

□□Topic□□□□□org.apache.rocketmq.tools.command.topic□□UpdateTopicSubCommand□□□□□Topic□□□□updateTopic□□□□□4-3□□□

□□□□4-3 updateTopic□□□

```
Option("b", "BrokerAddr", true, "create topic to which Broker");
Option("c", "ClusterName", true, "create topic to which Cluster");
Option("t", "topic", true, "topic name");
Option("r", "readQueueNums", true, "set read queue nums");
Option("w", "writeQueueNums", true, "set write queue nums");
Option("p", "perm", true, "set topic's permission(2|4|6), intro[2:W 4:R; 6:RW]");
Option("o", "order", true, "set topic's order(true|false");
Option("u", "unit", true, "is unit topic (true|false");
Option("s", "hasUnitSub", true, "has unit sub (true|false");
```

□□b□c□□□□□□□□□□□□□□□□□□□□-b□□□□b□□□□□□□Broker□□□□Topic□Message Queue□c□□□□□□□Cluster□□□□□Master Broker□□□□□Topic□Message Queue□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□4-9□□□

□□□□4-4 updateTopic□□□

```
CreateTopicRequestHeader requestHeader = new CreateTopicRequestHeader();
requestHeader.setTopic(topicConfig.getTopicName());
requestHeader.setDefaultTopic(defaultTopic);
requestHeader.setReadQueueNums(topicConfig.getReadQueueNums());
requestHeader.setWriteQueueNums(topicConfig.getWriteQueueNums());
requestHeader.setPerm(topicConfig.getPerm());
requestHeader.setTopicFilterType(topicConfig.getTopicFilterType().name());
requestHeader.setTopicSysFlag(topicConfig.getTopicSysFlag());
requestHeader.setOrder(topicConfig.isOrder());

RemotingCommand request = RemotingCommand.createRequestCommand(RequestCode.
UPDATE_AND_CREATE_TOPIC, requestHeader)
```

□□Topic□□□□□□□□□□Broker□Broker□□□□Topic□□□□□□□□□□□□□□□□□□□□□□4-5□□□

□□□□4-5 Broker□□updateTopic□□

```
    private RemotingCommand updateAndCreateTopic(ChannelHandlerContext ctx,
RemotingCommand request) throws RemotingCommandException {
     ...
this.BrokerController.getTopicConfigManager().updateTopicConfig(topicConfig);  //
[illegible]topicConfig
        this.BrokerController.registerBrokerAll(false, true); //[illegible]NameServer[illegible]
registerBroker[illegible]
        return null;
    }
```

[illegible]NameServer[illegible]NameServer[illegible]Topic[illegible]Topic[illegible]org.apache.rocketmq.namesrv.routeinfo[illegible]RouteInfoManager[illegible]registerBroker[illegible]Broker[illegible]Master[illegible]Broker[illegible]QueueData[illegible]Topic[illegible]QueueData[illegible]Topic[illegible]QueueData[illegible]QueueData[illegible]

4.2.2 □□□□ZooKeeper

ZooKeeper□Apache□□□□□□□□□□□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□ZooKeeper□□□□□□□□□□□Master□□□□RocketMQ□□□□□□□□□□□□□□Master□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

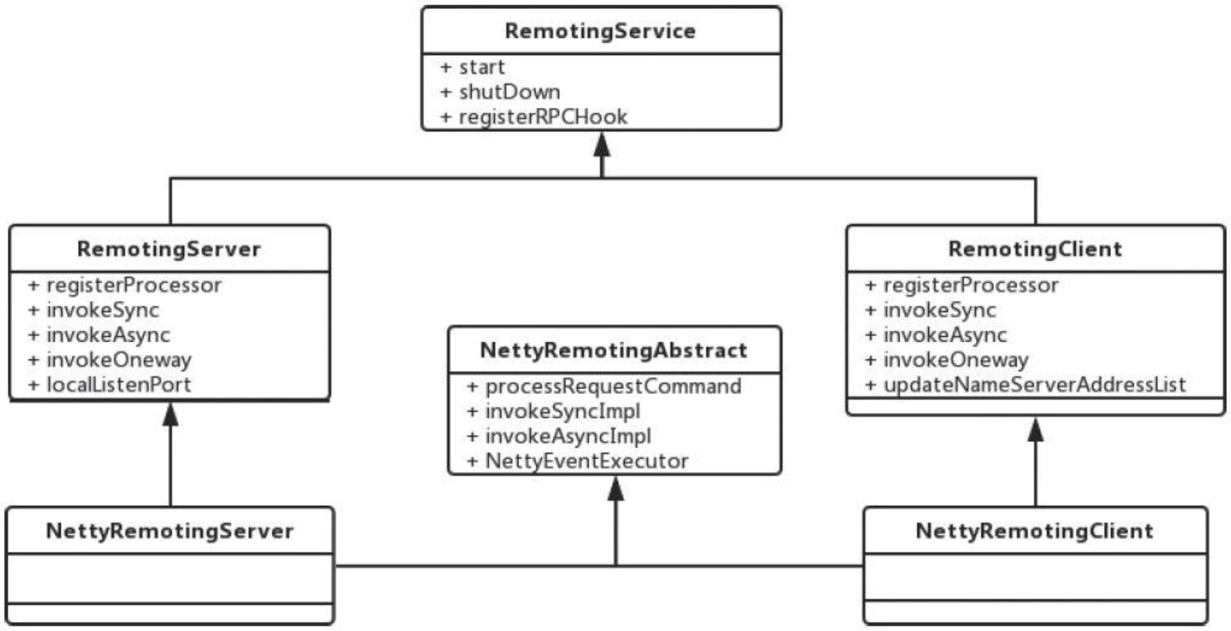
□□□□□□□□□□□□RocketMQ□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

4.3 [illegible]

[illegible]Socket[illegible]TCP[illegible]RocketMQ[illegible]

4.3.1 Remoting□□

RocketMQ□□□□□□□□Remoting□□□□□□□□□□□□□□□4-1□□□



□4-1　Remoting□□□□□□□□

RemotingService□□□□□□□□□□□□□□□

·void start□□□

·void shutdown□□□

·void registerRPCHook□RPCHook rpcHook□□

RemotingClient□RemotingServer□□RemotingService□□□□□□□□□□□□□□□RemotingClient□□□□□□□□□□□□4-6□□□

□□□□4-6　RemotingClient□□□□□□

```
void registerProcessor(final int requestCode, final NettyRequestProcessor
processor,final ExecutorService executor);
RemotingCommand invokeSync(final String addr, final RemotingCommand request, final
long timeoutMillis);
```

```
void invokeAsync(final String addr, final RemotingCommand request, final long
timeoutMillis,final InvokeCallback invokeCallback);
void invokeOneway(final String addr, final RemotingCommand request, final long
timeoutMillis);
void updateNameServerAddressList(final List<String> addrs);
```

□□□□□□□□□□□NettyRemotingClient□NettyRemotingServer□□□□□RemotingClient□RemotingServer□□□□□□□NettyRemoting-Abstract□□

□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□□□□□RemotingCommand□□□□□□□□□□□□□□□□□□□□

□□NameServer□□□□NameServerController□□□remotingServer□□□NameServer□□□□□□□□□□□□□□□□remotingServer□□□□□NameServer□□□□□□RemotingCommand□□□□□□□□□4-7□□□

□□□□4-7　NameServer□□□□□□□

```
@Override
public RemotingCommand processRequest(ChannelHandlerContext ctx, RemotingCommand
request) throws RemotingCommandException {
    if (log.isDebugEnabled()) {
        log.debug("receive request, {} {} {}",
            request.getCode(),
            RemotingHelper.parseChannelRemoteAddr(ctx.channel()),
            request);
    }
    switch (request.getCode()) {
        case RequestCode.PUT_KV_CONFIG:
            return this.putKVConfig(ctx, request);
        case RequestCode.GET_KV_CONFIG:
            return this.getKVConfig(ctx, request);
        case RequestCode.DELETE_KV_CONFIG:
            return this.deleteKVConfig(ctx, request);
        case RequestCode.REGISTER_BROKER:
            Version brokerVersion = MQVersion.value2Version(request.getVersion());
            if (brokerVersion.ordinal() >= MQVersion.Version.V3_0_11.ordinal()) {
                return this.registerBrokerWithFilterServer(ctx, request);
            } else {
                return this.registerBroker(ctx, request);
            }
        case RequestCode.UNREGISTER_BROKER:
            return this.unregisterBroker(ctx, request);
        case RequestCode.GET_ROUTEINTO_BY_TOPIC:
            return this.getRouteInfoByTopic(ctx, request);
        case RequestCode.GET_BROKER_CLUSTER_INFO:
            return this.getBrokerClusterInfo(ctx, request);
        case RequestCode.WIPE_WRITE_PERM_OF_BROKER:
            return this.wipeWritePermOfBroker(ctx, request);
        case RequestCode.GET_ALL_TOPIC_LIST_FROM_NAMESERVER:
            return getAllTopicListFromNameserver(ctx, request);
        case RequestCode.DELETE_TOPIC_IN_NAMESRV:
            return deleteTopicInNamesrv(ctx, request);
        case RequestCode.GET_KVLIST_BY_NAMESPACE:
            return this.getKVListByNamespace(ctx, request);
        case RequestCode.GET_TOPICS_BY_CLUSTER:
            return this.getTopicsByCluster(ctx, request);
        case RequestCode.GET_SYSTEM_TOPIC_LIST_FROM_NS:
            return this.getSystemTopicListFromNs(ctx, request);
        case RequestCode.GET_UNIT_TOPIC_LIST:
            return this.getUnitTopicList(ctx, request);
        case RequestCode.GET_HAS_UNIT_SUB_TOPIC_LIST:
            return this.getHasUnitSubTopicList(ctx, request);
        case RequestCode.GET_HAS_UNIT_SUB_UNUNIT_TOPIC_LIST:
            return this.getHasUnitSubUnUnitTopicList(ctx, request);
        case RequestCode.UPDATE_NAMESRV_CONFIG:
            return this.updateConfig(ctx, request);
        case RequestCode.GET_NAMESRV_CONFIG:
            return this.getConfig(ctx, request);
        default:
            break;
    }
    return null;
}
```

□Consumer□□□□□□□□□□□□□□□□□□□□□□RemotingCommand□□□□□□response□□□RemotingCommand□□□□□□□□4-8□□□

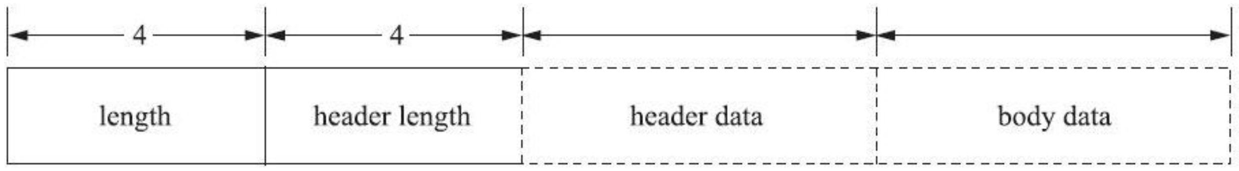
□□□□4-8　Consumer□□□□□□□□□□

```
    private PullResult pullMessageSync(//
        final String addr, // 1
        final RemotingCommand request, // 2
        final long timeoutMillis// 3
    ) throws RemotingException, InterruptedException, MQBrokerException {
        RemotingCommand response = this.remotingClient.invokeSync(addr, request,
    timeoutMillis);
        assert response != null;
        return this.processPullResponse(response);
    }
```

□□□□□□□□□RocketMQ□□□□□□□□□□RemotingCommand□□□□□□□□□□□□□□□□□□□□□□□Command□□□□□

4.3.2 □□□□□□□□

RocketMQ□□4-2□□□



□4-2 RocketMQ□□□□□

1□□□□□□□□4□□□□□□□□□□□□□□□□□□□□□□□

2□□□□□□□□4□□□□□□□□□□□□□□□□□

3□□□□□□□□Json□□□□□□□

4□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□RemotingCommand□decode□□□□□□□□□4-9□□□

□□□□4-9 □□□□□□

```
public static RemotingCommand decode(final ByteBuffer byteBuffer) {
    int length = byteBuffer.limit();
    int oriHeaderLen = byteBuffer.getInt();
    int headerLength = getHeaderLength(oriHeaderLen);
    byte[] headerData = new byte[headerLength];
    byteBuffer.get(headerData);
    RemotingCommand cmd = headerDecode(headerData, getProtocolType
(oriHeaderLen));
    int bodyLength = length - 4 - headerLength;
    byte[] bodyData = null;
    if (bodyLength > 0) {
        bodyData = new byte[bodyLength];
        byteBuffer.get(bodyData);
    }
    cmd.body = bodyData;
    return cmd;
}
```

□□□□□□□□□□RemotingCommand□encode□□□□□□□□□4-10□□□

□□□□□4-10　□□□□□□

```
public ByteBuffer encode() {
    // 1> header length size
    int length = 4;
    // 2> header data length
    byte[] headerData = this.headerEncode();
    length += headerData.length;
    // 3> body data length
    if (this.body != null) {
        length += body.length;
    }
    ByteBuffer result = ByteBuffer.allocate(4 + length);
    // length
    result.putInt(length);
    // header length
    result.put(markProtocolType(headerData.length, serializeTypeCurrentRPC));
    // header data
    result.put(headerData);
    // body data;
    if (this.body != null) {
        result.put(this.body);
    }
    result.flip();
    return result;
}
```

4.3.3 Netty□

RocketMQ□□□Netty□□□□RemotingServer□RemotingClient□□□□□□□□□Netty□□□□□□□□□□□□□□□□□□Java Socket□NIO□□□□□□□□□□□□Netty□□□□□□□□“□□□□□□+□□□□□□”□□□Server□Client□□□□□□□Netty□□□□□□□□□□□□□EventLoopGroup□Channel□Handler□□□□□□□□□□□□RocketMQ□□Netty□□□□□□□NettyRemotingServer□NettyRemotingClient□□□□□□□□□□□□□□□□□□□□Netty□

4.4 [illegible]

[illegible]NameServer[illegible]NameServer[illegible]RocketMQ[illegible]Producer[illegible]Consumer[illegible]NameServer[illegible]NameServer[illegible]RocketMQ[illegible]Netty[illegible]Netty[illegible]13[illegible]

第5章　消息队列的核心机制

Broker是RocketMQ的核心，大部分“重量级”工作都是由Broker完成的，包括接收Producer发过来的消息、处理Consumer的消费消息请求、消息的持久化存储、消息的HA机制以及服务端过滤功能等。

5.1 [illegible]

[illegible]

[illegible]600MB/s[illegible]100KB/s[illegible]6000[illegible]

[illegible]Linux[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

1[illegible]read[illegible]file[illegible]tmp_buf[illegible]len[illegible]

2[illegible]write[illegible]socket[illegible]tmp_buf[illegible]len[illegible]

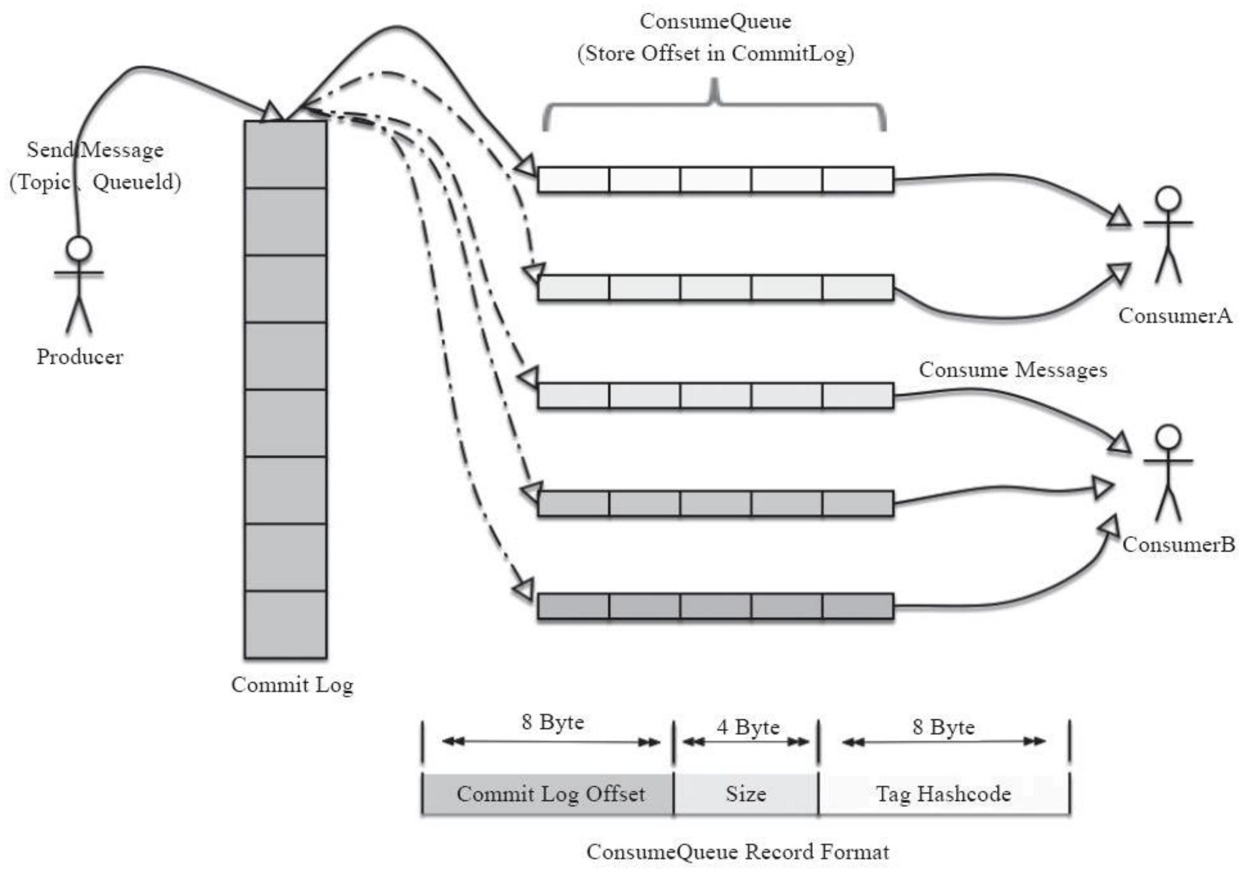
tmp_buf[illegible]4[illegible]read[illegible]file[illegible]tmp_buf[illegible]len[illegible]write[illegible]socket[illegible]tmp_buf[illegible]len[illegible]

[illegible]mmap[illegible]Java[illegible]MappedByteBuffer[illegible]Java 7[illegible]https://docs.oracle.com/javase/7/docs/api/java/nio/MappedByteBuffer.html [illegible]RocketMQ[illegible]“[illegible]”[illegible]

5.2 [illegible]

RocketMQ[illegible]5-1[illegible]

RocketMQ[illegible]ConsumeQueue[illegible]CommitLog[illegible]CommitLog[illegible]ConsumeQueue[illegible]Topic[illegible]Message Queue[illegible]ConsumeQueue[illegible]${$storeRoot}\consumequeue\${topicName}\${queueId}\${fileName}[illegible]



[illegible]5-1 RocketMQ[illegible]

CommitLog[illegible]Broker[illegible]CommitLog[illegible]ConsumeQueue[illegible]

${user.home}\store\${commitlog}\${fileName}□□
CommitLog□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□
CommitLog□□□□□□□□□□□ConsumeQueue□□□□□□□□□□□□□□□□□

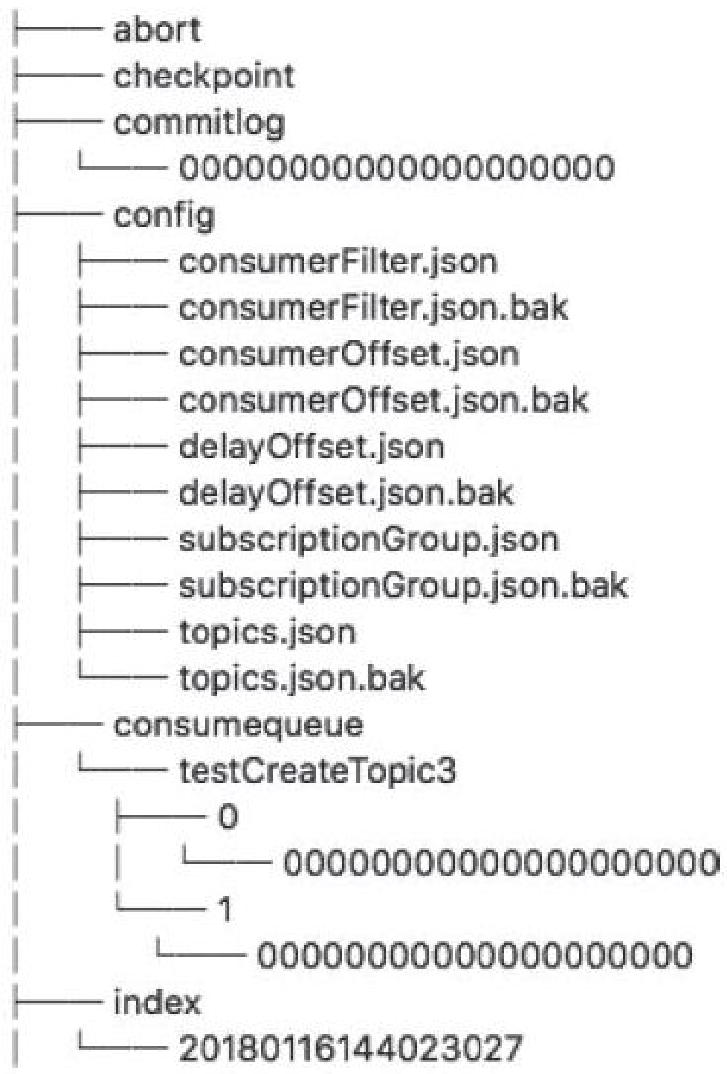
□□□□□□□□□□□□□□□□

1□CommitLog□□□□□□□□□□□□□□□

2□□□□□□□□□□□□□□□□pagecache□□□□□□□□□□□□□□□□cache
□□□□□□□□□□□□□□□□

3□□□□□□□□□□□□□ConsumeQueue□□□□□□□□□
ConsumeQueue□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
ConsumeQueue□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□CommitLog□ConsumeQueue□□□□□CommitLog□□□□
Consume Queues□Message Key□Tag□□□□□□□□
ConsumeQueue□□□□□□□□commitLog□□□□□□□

□□5-2□□□□□Broker□□□□□□□□□□□□□□□□□□□□commitlog□□□□
consumequeue□□□□□□□config□□□□Topic□Consumer□□□□□□□□
□□□□□□index□□□□□□□□□□□□□□□□□□□□□□□□



```
├── abort
├── checkpoint
├── commitlog
│   └── 00000000000000000000
├── config
│   ├── consumerFilter.json
│   ├── consumerFilter.json.bak
│   ├── consumerOffset.json
│   ├── consumerOffset.json.bak
│   ├── delayOffset.json
│   ├── delayOffset.json.bak
│   ├── subscriptionGroup.json
│   ├── subscriptionGroup.json.bak
│   ├── topics.json
│   └── topics.json.bak
├── consumequeue
│   └── testCreateTopic3
│       ├── 0
│       │   └── 00000000000000000000
│       └── 1
│           └── 00000000000000000000
├── index
│   └── 20180116144023027
```

□5-2　RocketMQ□Broker□□□□□□□□□□□□□

5.3 □□□□□□

RocketMQ□□□□□□□□Master□Slave□□□□□□□□□□□□□□□□Master□Slave□□□□□Broker□□□□□□□□□brokerId□□□□0□□□Broker□Master□□□0□□□□Broker□Slave□□□brokerRole□□□□□□□□Broker□Master□□Slave□Master□□□Broker□□□□□□Slave□□□Broker□□□□□□□□Producer□□□Master□□□Broker□□□□□□□Consumer□□□□Master□□□Broker□□□□□□Slave□□□Broker□□□□□□

□Consumer□□□□□□□□□□□□□□□Master□□□□Slave□□□Master□□□□□□□□□□□Consumer□□□□□□□□Slave□□□□□□□□Consumer□□□□□□□□Master□□□□□□□□□□□Consumer□□□□□Slave□□□□□□□□Consumer□□□□□□□□□□□□□□□□□

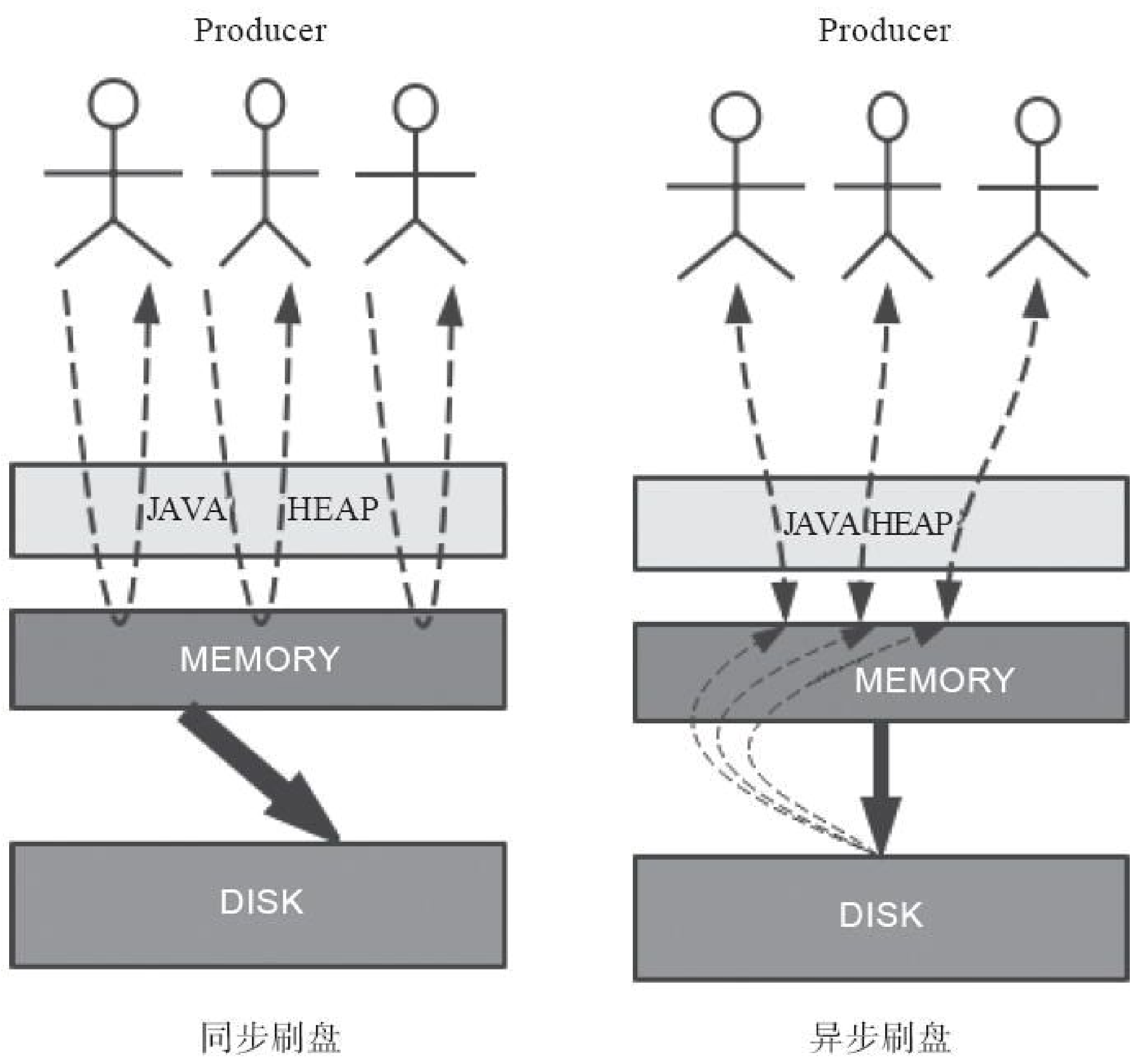
□□□□□□□□□□□□□□□□□Topic□□□□□Topic□□□Message Queue□□□□□Broker□□□□□Broker□□□□□brokerId□□□□□□□Broker□□□□□□□□Broker□□Master□□□□□□□□□Master□□□□□Producer□□□□□□□□□RocketMQ□□□□□□□Slave□□□□Master□□□□□□□□□□□□□Slave□□Master□□□□□□□Slave□□□Broker□□□□□□□□□□□□□□□□□Broker□

5.4 [illegible]

RocketMQ[illegible]RocketMQ[illegible]Producer[illegible]RocketMQ[illegible]

·[illegible]PAGECACHE[illegible]

·[illegible]PAGECACHE[illegible]



[illegible]5-3 [illegible]

[illegible]Broker[illegible]flushDiskType[illegible]SYNC_FLUSH[illegible]ASYNC_FLUSH[illegible]

5.5 □□□□□□□□□

□□□□Broker□□Master□Slave□□□□□□Master□□□Slave□□□□□□□□□□□□□□□□□□□□□□□Master□Slave□□□□□□□□□□□□□□□□□□□□□□□□□□□Master□□□□□□□□□□□□□□□□□

□□Master□□□□□□□□□□□□□□□□Slave□□□□□□□□□□□□□□□□□□□Master□□□□Slave□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□Broker□□□□□□brokerRole□□□□□□□□□□□□□□□□□□ASYNC_MASTER□SYNC_MASTER□SLAVE□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□SYNC_FLUSH□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Master□Save□□□ASYNC_FLUSH□□□□□□□□□□□□□SYNC_MASTER□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5.6 [illegible]

[illegible]RocketMQ[illegible]“[illegible]”[illegible]RocketMQ[illegible]“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]RocketMQ[illegible]

□6□　□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

6.1 [illegible]

[illegible]3[illegible]Topic[illegible]ID[illegible]

6.1.1 [illegible]

RocketMQ[illegible]Topic[illegible]Consumer[illegible]Consumer[illegible]Consumer[illegible]

[illegible]Topic[illegible]Producer[illegible]Consumer[illegible]Topic[illegible]

[illegible]RocketMQ[illegible]

6.1.2 □□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ID□□□□□□□□□Message Queue□□□□□□□□□□□□□□□Message Queue□□□□□□□□□□□□□□□□□□□□□□□

□□□□□MessageQueueSelector□□□□□□□□□□□Message Queue□□□□□□6-1□□□

□□□□6-1 MessageQueueSelector□□

```
for (int i = 0; i < 100; i++) {
    int orderId = i;
    //Create a message instance, specifying topic, tag and message body.
    Message msg = new Message("OrderTopic8", tags, "KEY" + i,
        ("Hello RocketMQ " +orderId+"  "+
i).getBytes(RemotingHelper.DEFAULT_CHARSET));
    SendResult sendResult = Producer.send(msg, new MessageQueueSelector() {
        @Override
        public MessageQueue select(List<MessageQueue> mqs, Message msg, Object
arg) {
            System.out.println("queue selector mq nums:"+mqs.size());
            System.out.println("msg info:"+msg.toString());
            for(MessageQueue mq: mqs){
                System.out.println(mq.toString());
            }
            Integer id = (Integer) arg;
            int index = id % mqs.size();
            return mqs.get(index);
        }
    }, orderId);
    System.out.println(sendResult);
}
```

□□□□□□□MessageListenerOrderly□□□□□Message Queue□□□□□□□□□□□□□□□□□6-2□□□

□□□□6-2 MessageListenerOrderly□□

```
consumer.registerMessageListener(new MessageListenerOrderly() {
    AtomicLong consumeTimes = new AtomicLong(0);
    @Override
    public ConsumeOrderlyStatus consumeMessage(List<MessageExt> msgs,
                            ConsumeOrderlyContext context) {
        System.out.printf(" Received New Messages: " + new
String(msgs.get(0).getBody()) + "%n");
            return ConsumeOrderlyStatus.SUCCESS;
        }
    });
```

Consumer[illegible]MessageListenerOrderly[illegible]Consumer[illegible]setConsumeThreadMin[illegible]setConsumeThreadMax[illegible]setPull-BatchSize[illegible]setConsumeMessageBatchMaxSize[illegible]Consumer[illegible]PullBatchSize[illegible]Broker[illegible]Message Queue[illegible]32[illegible]ConsumeMessageBatchMaxSize[illegible]Consumer[illegible]Executor[illegible]MessageListener[illegible]List<MessageExt>msgs[illegible]1[illegible]

[illegible]MessageListenerOrderly[illegible]MessageListenerOrderly[illegible]Consumer Queue[illegible]Consumer Queue[illegible]Consumer Queue[illegible]Consumer Queue[illegible]

6.2 □□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□”□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□Producer□□□□setRetryTimesWhenSendFailed□□□□□□□□□□□□□□□□□□□□□2□□□□□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

6.3 [illegible]

[illegible]

6.3.1 □□□□NameServer

NameServer□RocketMQ□□□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Broker□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□Producer□Consumer□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□NameServer□NameServer□□□□□□□□□□□Broker□□□□□□□□□□□□□NameServer□□□□□□□NameServer□□□□□□□□□□□□□

□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□

1□□□□□□□□□□□□Producer□□□□□Producer.setNamesrvAddr□"name-server1-ip□port□name-server2-ip□port"□□□□□□mqadmin□□□□□□□□□□□-n name-server-ip1□port□name-server-ip2□port□□□□□□□□□□□□□□□□□□□□□□□□□□□defaultMQAdminExt.setNamesrvAddr□"name-server1-ip□port□name-server2-ip□port"□□□□□

2□□□Java□□□□□□□□□□□option□rocketmq.namesrv.addr□

3□□□Linux□□□□□□□□□□□□□□□□□□NAMESRV_ADDR□

4□□□HTTP□□□□□□□□□□□□□□□□□□□□□□□□□□HTTP□□□□□□□□□□NameServer□□□□□□□URL□http://jmenv.tbsite.net:8080/rocketmq/nsaddr □□□□□□□□□□□□□□rocketmq.namesrv.domain□□□□□jmenv.tbsite.net□□□rocketmq.namesrv.domain.subgroup□□□□□nsaddr□

□4□□□□□□□□□□□□□□□□□□□□□□NameServer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2□□□□□□□□URL□□□□□□□NameServer□□□

6.3.2 □□□□Broker

□□□□□□□□□□□□□□□□□□□□□□□□□□Broker□□□□□□□□□Broker□□□□□□Topic□□□□□□□□□□□Topic□□□□□□□□□□□□□□Broker□□□□

□□□□□□□□□□□□□□Topic□□□□□Broker□□□□□□□□□□□□□□□□□□□updateTopic□□□□□□□Topic□□□□□□□Broker□□□□□□□□□□TestTopic□□□□□□Topic□□□□□□□□□□□□□□□□□□Broker□□□□□192.168.0.1□10911□□□□□□□□□□□□□sh./bin/mqadmin updateTopic-b 192.168.0.1□10911-t TestTopic-n 192.168.0.100□9876□□□□□□□□Broker□□□□□TestTopic□□□□8□□□□□□

□□□□□□□□□□□□□□□□□□Broker□□□□□□□□□□□□Broker□□□□□□□□□□Producer□□□□Topic□□□□Master Broker□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□Broker□□□□□□□□□

□□□Topic□□□Master Broker□□□□□□□□□□□□□□□□□□□□□□□Producer□□□□□□□□□□□□□□□□□□□□□send□msg□□□□□DefaultMQProducer□□□□□□□□□□□□□□□Broker□□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□send□msg□callback□□□□□sendOneWay□□□□□□□□□□□□□□□□□□□□□sendOneWay□□□□□□□□□Producer.setRetryTimesWhenSendFailed□□□□□□□□□□□□□□□□DefaultMQProducer□□□30□□NameServer□□□□□□□□□□Producer□□□□□□□□□□Broker□□□□□□□□□□□□□□□□□□□□□□□□

□□Producer□□□□□□□□□□□Master□□□Slave□□□□□□□□□□□□□□□□Producer□□□Master Broker□□□□□□□□□□□□□□□□Slave□□□□□□□□□□□□□Master Broker□□□□□□□□□□□□□□□□□□Master Broker□□□□□□Producer□□□□□□□□□

□Linux□kill pid□□□□□□□□□□Broker□BrokerController□□□shutdown□□□□□□□□□□□ShutdownHook□□□□Linux□kill□□□□□□□kill-9□□shutdown□□□□□□□□□□□□□RocketMQ□□□□□□mqshutdown broker□□□□Broker□□□□□□□□□□□

6.4 [illegible]

[illegible]

1[illegible]Broker[illegible]

2[illegible]Broker[illegible]Crash[illegible]

3[illegible]OS Crash[illegible]

4[illegible]

5[illegible]

6[illegible]CPU[illegible]

[illegible]RocketMQ[illegible]Topic[illegible]Master[illegible]Broker[illegible]Master[illegible]Slave[illegible]

[illegible]1[illegible]Consumer[illegible]Master[illegible]Consumer[illegible]Slave[illegible]Master[illegible]Consumer[illegible]Master[illegible]Master[illegible]Consumer[illegible]Slave[illegible]Consumer[illegible]Consumer[illegible]Master[illegible]Consumer[illegible]Consumer[illegible]Master[illegible]offset[illegible]

[illegible]1[illegible]Producer[illegible]Master[illegible]Producer[illegible]Topic[illegible]Master[illegible]Producer[illegible]

[illegible]2[illegible]3[illegible]4[illegible]Master[illegible]Slave[illegible]SYNC_FLUSH[illegible]1[illegible]

□5□6□□□□□□□□□□□□□5□6□□□□□□□□□□□□□□□□□□□□□□□□Master□Slave□□□□□□□□□□□□□□□□□□□□5□6□□□□□□□□□□□□□□□□□□□□Master□Slave□□□□□□□□□□□Sync□□□□□□□□

□□□□□□□□□□

1□□Master□□□Master□□Slave□

2□□□□□□□□SYNC_MASTER□

3□Producer□□□□□□□

4□□□□□□□□SYNC_FLUSH□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

6.5 [illegible]

[illegible]RocketMQ[illegible]Topic[illegible]

[illegible]Topic[illegible]AA[illegible]AB[illegible]AC[illegible]AB[illegible]AC[illegible]AB[illegible]AC[illegible]AA[illegible]AB[illegible]AC[illegible]

[illegible]AA[illegible]Topic[illegible]AA[illegible]Topic[illegible]AB[illegible]AC[illegible]Topic[illegible]Topic[illegible]Consumer[illegible]Topic[illegible]AA[illegible]Topic[illegible]AB[illegible]AC[illegible]

[illegible]Topic[illegible]100[illegible]Producer[illegible]RocketMQ[illegible]Consumer[illegible]1[illegible]100[illegible]2[illegible]99[illegible]2[illegible]

[illegible]Topic[illegible]Topic[illegible]MessageQueue[illegible]100[illegible]Producer[illegible]MessageQueue[illegible]DefaultMQPushConsumer[illegible]Topic[illegible]MessageQueue[illegible]MessageQueue[illegible]

DefaultMQPushConsumer[illegible]pullBatchSize[illegible]32[illegible]MessageQueue[illegible]32[illegible]pullBatchSize[illegible]1[illegible]

[illegible]“[illegible]”[illegible]TypeA[illegible]TypeB[illegible]TypeC[illegible]TypeA[illegible]TypeA[illegible]TypeB[illegible]TypeC[illegible]

[illegible]Topic[illegible]PullConsumer[illegible]MessageQueue[illegible]
[illegible]Topic[illegible]Consumer[illegible]
Consumer[illegible]

6.6 [illegible]

[illegible]“[illegible]”[illegible]RocketMQ[illegible]

[illegible]7[illegible] [illegible]

[illegible]RocketMQ[illegible]

7.1 □Broker□□□□□□□

□Broker□□□□□□□□□□□□□□□□□□□Consumer□□□□□□□□□□□□□
□□□Broker□□□□□□□□□□□□□

7.1.1 [illegible]Tag[illegible]Key

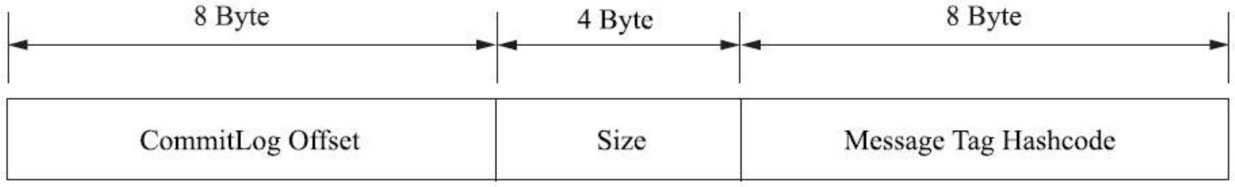
[illegible]Topic[illegible]Tag[illegible]Tag[illegible]Tag[illegible]Tag[illegible]Tag[illegible]Broker[illegible]

[illegible]Key[illegible]Key[illegible]Key[illegible]Key[illegible]Broker[illegible]Key[illegible]Key[illegible]

Tag[illegible]Key[illegible]Tag[illegible]Consumer[illegible]Key[illegible]

7.1.2 □□Tag□□□□

□Tag□□□□□□□□□□□□□□□□Tag□□□Tag□□□□□□□□□□□□□□□Message□□□□□□□□□Message□□□□□Tag□□□Tag□□□□□□□□□Broker□□□□ConsumeQueue□□□□□□□□□CommitLog□□□□□□□□□□□□□□□□ConsumerQueue□□□□□□□□7-1□□□



□7-1 ConsumerQueue□□□□□

Consume Queue□□□□□□□□□Tag□□□hashcode□□□□□□□□□□□□□Tag□□□□□□□□□□□□hashcode□□□hashcode□□□□□□□□□□□□CommitLog□□□□□□□□□Hash□□□□□□□□□□□□□□□□□□□Message Tag□□□□□□Hash□□□□□□□□

7.1.3 [illegible]SQL[illegible]

[illegible]Tag[illegible]Message[illegible]putUserProperty[illegible]7-1[illegible]

[illegible]7-1 [illegible]

```
Message msg = new Message("TopicTest",
    tag,
    ("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT_CHARSET)
);
// Set some properties.
msg.putUserProperty("a", String.valueOf(i));
msg.putUserProperty("b",  “hello”);
```

[illegible]a[illegible]b[illegible]SQL[illegible]PushConsumer[illegible]

```
DefaultMQPushConsumer consumer = new
DefaultMQPushConsumer("please_rename_unique_group_name_4");  // only subsribe
messages have property a, also a >=0 and a <= 3 consumer.subscribe("TopicTest",
MessageSelector.bySql("a between 0 and 3");
consumer.registerMessageListener(new MessageListenerConcurrently()
{     @Override     public ConsumeConcurrentlyStatus consumeMessage
    (List<MessageExt> msgs, ConsumeConcurrentlyContext context)
{         return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;     } });
consumer.start();
```

[illegible]SQL[illegible]

·[illegible]>[illegible]>=[illegible]<[illegible]<=[illegible]BETWEEN[illegible]=[illegible]

·[illegible]=[illegible]<>[illegible]IN[illegible]

·IS NULL or IS NOT NULL[illegible]

·[illegible]AND[illegible]OR[illegible]NOT[illegible]

[illegible]

·□□□□□□123□3.1415□

·□□□□□□'abc'□□□□□□□□□□

·NULL□□□□□□□□

·□□□□TRUEorFALSE□

SQL□□□□□□□□□□Broker□□□□□□□□□□□□□□□SQL□□□□□□□□□□□□Tag□□□□□

7.1.4 Filter Server[illegible]

Filter Server[illegible]SQL[illegible]Java[illegible]Java[illegible]

[illegible]Filter Server[illegible]Broker[illegible]filterServer-Nums=3[illegible]Broker[illegible]3[illegible]Filter Server[illegible]Filter Server[illegible]RocketMQ[illegible]Consumer[illegible]Broker[illegible]Java[illegible]Consumer[illegible]Broker[illegible]CPU[illegible]java[illegible]Broker[illegible]7-2[illegible]

[illegible]7-2 [illegible]

```
public class MessageFilterImpl implements MessageFilter {
    @Override
    public boolean match(MessageExt msg) {
        String property = msg.getUserProperty("SequenceId");
        if (property != null) {
            int id = Integer.parseInt(property);
            if ((id % 3) == 0 && (id > 10)) {
                return true;
            }
        }
        return false;
    }
}
```

[illegible]“SequenceId”[illegible]7-3[illegible]

[illegible]7-3 [illegible]FilterServer[illegible]Consumer[illegible]

```
public static void main(String[] args) throws InterruptedException,
MQClientException {
    DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("Consumer-
GroupNamecc4");
    // [illegible]Java[illegible]
    String filterCode = MixAll.file2String("/home/admin/MessageFilterImpl.java");
        consumer.subscribe("TopicFilter7",
"com.alibaba.rocketmq.example.filter.MessageFilterImpl", filterCode);
        consumer.registerMessageListener(new MessageListenerConcurrently() {

            @Override
            public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
msgs,
                ConsumeConcurrentlyContext context) {
            System.out.println(Thread.currentThread().getName() + " Receive New
Messages: " + msgs);
                return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
            }
        });
        consumer.start();
        System.out.println("Consumer Started.");
    }
```

□□□Filter Server□Consumer□□□□□□□□□□□□□□□□□□□□Broker□□Broker□□Filter Server□□□□□□□□□□match□□□□□□□□□□

7.2 [illegible]Consumer[illegible]

[illegible]Consumer[illegible]Consumer[illegible]

[illegible]1[illegible]

[illegible]ConsumerGroup[illegible]Clustering[illegible]Consumer[illegible]Consumer[illegible]Consumer[illegible]Consumer[illegible]Topic[illegible]Read Queue[illegible]Consumer[illegible]Consumer[illegible]Consumer[illegible]consumeThreadMin[illegible]consumeThreadMax[illegible]

[illegible]2[illegible]

[illegible]update[illegible]update10[illegible]update1[illegible]Consumer[illegible]consumeMessageBatchMaxSize[illegible]1[illegible]N[illegible]N[illegible]

[illegible]3[illegible]

Consumer[illegible]Consumer[illegible]Producer[illegible]7-4[illegible]

[illegible]7-4 [illegible]

```
public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt> msgs,
ConsumeConcurrentlyContext context) {
long Offset = msgs.get(0).getQueueOffset();
String maxOffset = msgs.get(0).getProperty(Message.PROPERTY_MAX_OFFSET);     long
diff = Long.parseLong(maxOffset) - Offset;
if (diff > 90000) {
return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
}
//□□□□□
 return ConsumeConcurrentlyStatus.CONSUME_SUCCESS; }
```

□□□□□□□□□□□□□□□□□□90000□□□□□□□□□□□□□□□□□□□□□□□□

7.3 Consumer[illegible]

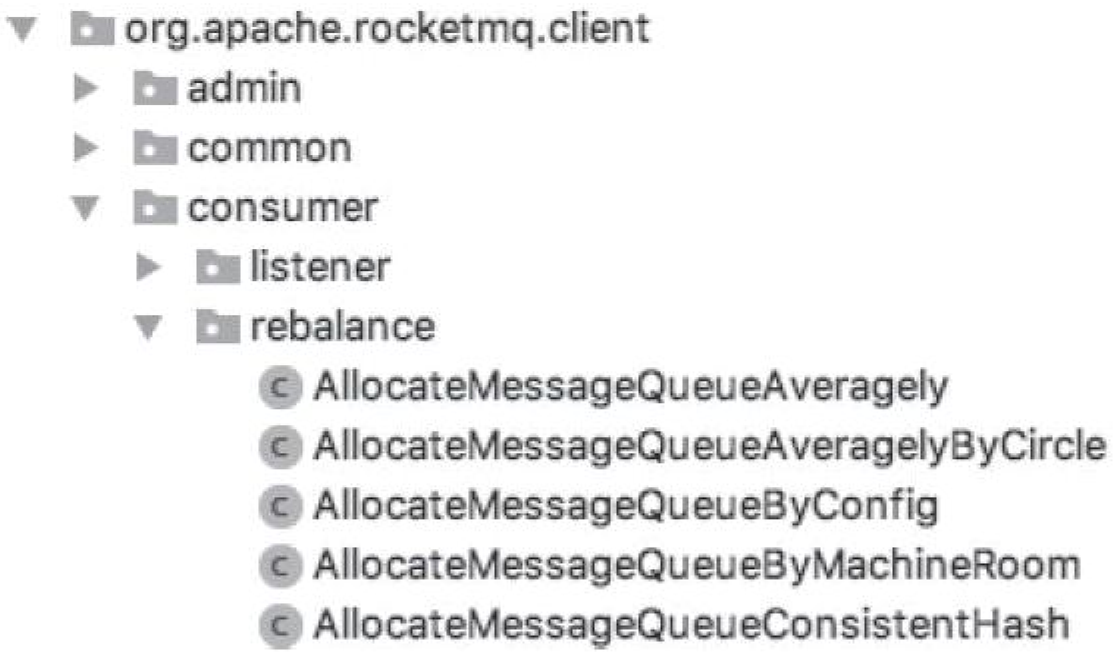
[illegible]Consumer[illegible]Consumer[illegible]Consumer[illegible]Consumer[illegible]

[illegible]ConsumerGroup[illegible]Consumer[illegible]Consumer[illegible]RocketMQ[illegible]Consumer[illegible]Consumer[illegible]Broker[illegible]

7.3.1 DefaultMQPushConsumer□□□□□

DefaultMQPushConsumer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□DefaultMQPushConsumer□□□□□□□□□□□□doRebalance□□□□□□□□□ConsumerGroup□□□□□DefaultMQPush-Consumer□□□□Consumer□□□□□doRebalance□□□

□□7-2□□□□□□□□□□□□□□□□□□□□□□□□AllocateMessageQueueAveragely□□□□□□□□□Topic□Message Queue□□□□□ConsumerGroup□□Consumer□□□□□□□□□□□□□□□□□Message Queue□□Topic□□□□Message Queue□□□□□□Consumer□□□□Message Queue□Consumer□□□□□□□□□□□□□□□□□□□□□


```
org.apache.rocketmq.client
    admin
    common
    consumer
        listener
        rebalance
            AllocateMessageQueueAveragely
            AllocateMessageQueueAveragelyByCircle
            AllocateMessageQueueByConfig
            AllocateMessageQueueByMachineRoom
            AllocateMessageQueueConsistentHash
```

□7-2 RocketMQ□□□□□□□□□

□AllocateMessageQueueAveragely□□□□□□□□Topic□□□□□Message Queue□□□3□□Consumer□□□2□□□□□□□Consumer□□□□Topic□□□□□□□□□□□□□□□□□□□□□□Consumer□□4□□□□□□□□Consumer□□□□□□□□□3□Consumer□□□Topic□□□□□□□□□□

Message Queue□□□□□□□□□□□□□□□□□□□□□□□□Topic□Message Queue□□□□16□

7.3.2 DefaultMQPullConsumer[illegible]

Pull Consumer[illegible]Message Queue[illegible]Message Queue[illegible]Offset[illegible]

DefaultMQPullConsumer[illegible]registerMessageQueueListener[illegible]7-5[illegible]

[illegible]7-5 registerMessageQueueListener

```
Consumer.registerMessageQueueListener("TOPICNAME", new MessageQueue-Listener() {
public void MessageQueueChanged(String Topic, Set<MessageQueue> mqAll,
Set<MessageQueue> mqDivided) }
```

registerMessageQueueListener[illegible]Consumer[illegible]MQPullConsumerScheduleService[illegible]Class[illegible]DefaultMQPushConsumer[illegible]Pull[illegible]7-6[illegible]

[illegible]7-6 [illegible]MQPullConsumerScheduleService[illegible]

```
public class PullConsumerServiceTest {
    public static void main(String[] args) throws MQClientException {
        final MQPullConsumerScheduleService scheduleService = new MQPull-
ConsumerScheduleService("PullConsumerService1");
        scheduleService.getDefaultMQPullConsumer().setNamesrvAddr("localh-
ost:9876");
        scheduleService.setMessageModel(MessageModel.CLUSTERING );
        scheduleService.registerPullTaskCallback("testPullConsumer", new
PullTaskCallback() {
            public void doPullTask(MessageQueue mq, PullTaskContext context) {
                MQPullConsumer Consumer = context.getPullConsumer();
                try {
                    long Offset = Consumer.fetchConsumeOffset(mq, false);
                    if (Offset < 0)
                        Offset = 0;
                    PullResult pullResult = Consumer.pull(mq, "*", Offset, 32);
                    System.out.printf("%s%n", Offset + "\t" + mq + "\t" +
pullResult);
                    switch (pullResult.getPullStatus()) {
                        case FOUND:
                            break;
                        case NO_MATCHED_MSG:
                            break;
                        case NO_NEW_MSG:
                        case OFFSET_ILLEGAL:
                            break;
                        default:
                            break;
                    }
                    Consumer.updateConsumeOffset(mq,
pullResult.getNextBeginOffset());
                    context.setPullNextDelayTimeMillis(1000);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
        scheduleService.start();
    }
}
```

□□□□□□□□MQPullConsumerScheduleService□□□□□□□□□□□□□□□□□□□□□7-7□□□

□□□□7-7　MQPullConsumerScheduleService□□□□□□□

```
class MessageQueueListenerImpl implements MessageQueueListener {
    @Override
    public void MessageQueueChanged(String Topic, Set<MessageQueue> mqAll,
Set<MessageQueue> mqDivided) {
        MessageModel MessageModel =

MQPullConsumerScheduleService.this.defaultMQPullConsumer.getMessageModel();
        switch (MessageModel) {
            case BROADCASTING:
                MQPullConsumerScheduleService.this.putTask(Topic, mqAll);
                break;
            case CLUSTERING :
                MQPullConsumerScheduleService.this.putTask(Topic, mqDivided);
                break;
            default:
                break;
        }
    }
}
```

□□□□□□□□□□□□□□□MessageQueueListenerImpl□□□□□□□□□□□□□□□

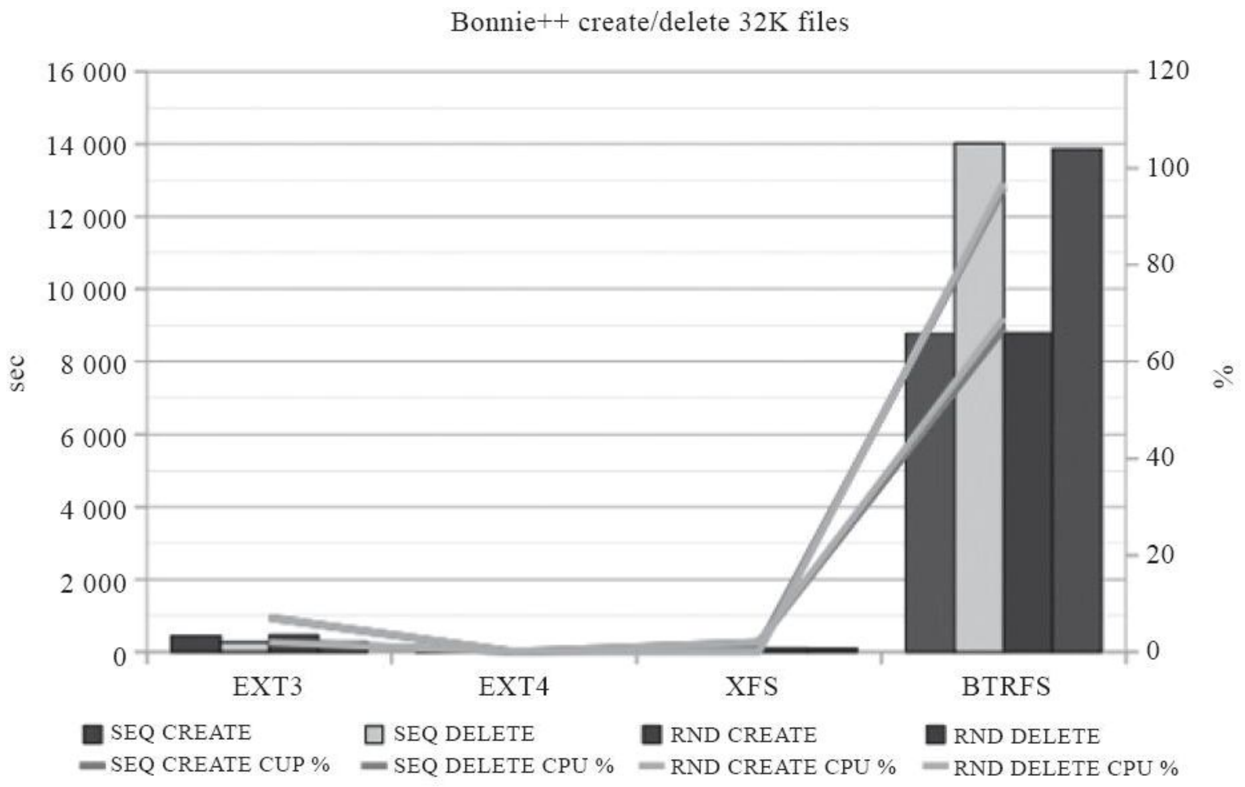
7.4 [illegible]Producer[illegible]

[illegible]Oneway[illegible]Oneway[illegible]Socket[illegible]

[illegible]Producer[illegible]Producer[illegible]Producer[illegible]RocketMQ[illegible]DirectMem[illegible]CommitLog[illegible]RocketMQ[illegible]HDD[illegible]SSD[illegible]90[illegible]+[illegible]TPS[illegible]

[illegible]Linux[illegible]EXT4[illegible]IO[illegible]deadline[illegible]

[illegible]7-3[illegible]EXT4[illegible]/[illegible]EXT3[illegible]RocketMQ[illegible]CommitLog[illegible]/[illegible]



□7-3　□□□□□□□Bonnie++□□□/□□32K□□□□□□□

□□□IO□□□□□□□□□□deadline□deadline□□□□□□□□□□□□□□□□□□□□□□□□read□write□□□□□□□□□□□□read□write□□□□□□read□write□□□□□□□□□□□□□□□□□□IO□□□□□□□□□□□□□IO□□IO□□□□□□□□□read□write□□IO□□□□□□□□□

7.5 [illegible]

[illegible]RocketMQ[illegible] Producer[illegible]Consumer[illegible]

[illegible]TPS[illegible]QPS[illegible]

[illegible]1[illegible]TOP[illegible]CPU[illegible]

```
Tasks: 109 total,   1 running, 108 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.2 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  8010440 total,  1556880 free,  1626048 used,  4827512 buff/cache
KiB Swap:    0 total,   0 free,   0 used.  6058356 avail Mem
```

[illegible]CPU[illegible]99.8%[illegible]8G[illegible]1.5G[illegible]

[illegible]2[illegible]Linux[illegible]sar[illegible]

```
#sar -n DEV 2 10
Average: IFACE rxpck/s txpck/s rxkB/s txkB/s rxcmp/s txcmp/s rxmcst/s
Average: eth0   6.03  6.18   1.39 0.99   0.00   0.00   0.00
Average: eth1   4.41   3.82   0.42   0.98   0.00   0.00   0.00
```

·IFACE[illegible]LAN[illegible]

·rxpck/s[illegible]

·txpck/s[illegible]

·rxbyt/s[illegible]

·txbyt/s[illegible]

·rxcmp/s[illegible]

·txcmp/s[illegible]

·rxmcst/s[illegible]

[illegible]iperf3[illegible]netstat–t[illegible]

[illegible]iostat[illegible]

```
#iostat -xdm 1
Linux 3.10.0-514.6.1.el7.x86_64 (iZ2zehfpu32ir7r3vlhhuwZ)   12/28/2017
    _x86_64_(4 CPU)

Device:   rrqm/s   wrqm/s  r/s  w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await
r_await w_await  svctm  %util
vda   0.00  1.04    0.01    1.15  0.00  0.01    19.84  0.00    2.45
    1.58    2.46   0.39   0.05
vdb   0.00  0.00    0.00    0.00  0.00  0.00    14.75  0.00    0.11
    0.11    0.00   0.09   0.00
```

[illegible]CPU[illegible]

[illegible]CPU[illegible]bug[illegible]

[illegible]Java[illegible]Java[illegible]profiling[illegible]jvisualvm[illegible]jstack[illegible]perfJ[illegible]

[illegible]Java[illegible]

7.6 [illegible]

[illegible]RocketMQ[illegible]Tag[illegible]SQL[illegible]FilterServer[illegible]

[illegible]Broker[illegible]Consumer[illegible]Producer[illegible]

□8□　□□□□□□□

8.1　□SpringBoot□□□RocketMQ

Spring Boot□□□□□□□□Java□□□□□□□□□□□□“Spring□□□□”□□□□□□□□□Spring□□□□□□□RocketMQ□

8.1.1 □□□□

□Spring Boot□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Maven □□□□□□□□pom.xml□□□RocketMQ□□□□□□□□□8-1□□□

□□□□8-1 Maven□□□RocketMQ□□

```
<dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-client</artifactId>
    <version>4.2.0</version>
</dependency>
```

□□□□□□□□□□□Spring Boot□□□□□RocketMQ□Producer□ Consumer□□□□

□□RocketMQ□□□□□□□□□□□□□□□□□application.properties□ □□□□□□□□□□□□□□□□@Value□□□□□□□□□□□□□□NameServer□□□□ GoupName□□□Topic□□□□□□□□□□□Producer□Consumer□□□□□□ □□properties□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□Producer□Consumer□□□□□□□□□□□□□ □□□□□□□□□□□Service□□□□□□□□□Producer□Consumer□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Object□□□□□□□□□□□ @PostConstruct□□□□□□□□□□□□□@PreDestroy□□□□Spring Boot □□□□Producer□□□□□□□□□8-2□□□

□□□□8-2 Spring Boot□□□□Producer□□

```
@Service
public class ProducerService {
private DefaultMQProducer producer = null;
    @PostConstruct
    public void initMQProducer() {
        producer = new DefaultMQProducer("producerGoupName");
        producer.setNamesrvAddr(metaqNameserver);
        producer.setRetryTimesWhenSendFailed(3);
        try {
            producer.start();
        } catch (MQClientException e) {
            e.printStackTrace();
        }
    }
    public void send(String topic, String msg) {
        Message msg = new Message(topic, "", "", msg.getBytes());

        try {
            producer.send(msg);
            return;
        } catch (Exception e) {
            e.printStackTrace();
        }
            return□
    }
    @PreDestroy
    public void shutDownProducer() {
        if (producer != null) {
            producer.shutdown();
        }
    }
}
```

□□Consumer□□□□□Producer□□□□□□□□□□□□□□□□□Class□□□□□□□□shutdown□□□□□□□□□□□□□□□□□□□□□□□□□

8.1.2 □□Spring Messaging□□□□

□□□□□□□□□□□□□□□□□□□□□□“Spring Style”□Spring Boot□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Kafka□RabbitMQ□RocketMQ□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□RocketMQ□□□□□□□□□□8-3□□□

□□□□8-3 Spring Boot□RocketMQ□□

```
<!--□pom.xml□□□□□-->
<dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>spring-boot-starter-rocketmq</artifactId>
    <version>1.0.0-SNAPSHOT</version>
</dependency>
```

□□mvn□□□□□□□□□□□GitHub□□□□□□□□□□□

□□□□properties□□□□□□□□□□□□□□□8-4□□□

□□□□8-4 Spring Boot□RocketMQ□□□□□□

```
## application.properties
spring.rocketmq.name-server=127.0.0.1:9876
spring.rocketmq.producer.group=my-group
spring.rocketmq.producer.retry-times-when-send-async-failed=0
spring.rocketmq.producer.send-msg-timeout=300000
spring.rocketmq.producer.compress-msg-body-over-howmuch=4096
spring.rocketmq.producer.max-message-size=4194304
spring.rocketmq.producer.retry-another-broker-when-not-store-ok=false
spring.rocketmq.producer.retry-times-when-send-failed=2
```

□□□□□□□□□□□□□□□□□□□□Spring Boot□□□RocketMQ□□□□□□□□□□□□Spring Messaging□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Spring Boot□□□Messaging□□□□□GitHub□□rocketmq-externals□□□

8.2 [illegible]RocketMQ

[illegible]RocketMQ[illegible]MQ[illegible]RocketMQ[illegible]RocketMQ[illegible]

[illegible]

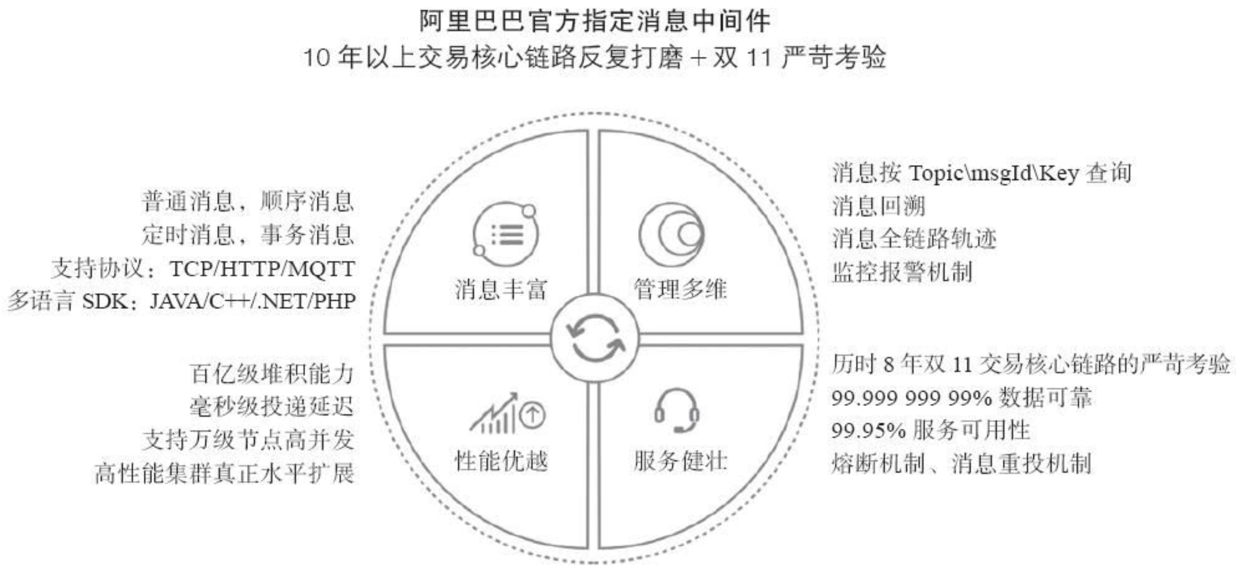
[illegible]MQ[illegible]MQ[illegible]Demo[illegible]8-5[illegible]

[illegible]8-5 [illegible]MQ[illegible]

```
public class ProducerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        // [illegible] MQ [illegible] Producer ID
        properties.put(PropertyKeyConst.ProducerId, "XXX");
        // [illegible] AccessKey[illegible]
        properties.put(PropertyKeyConst.AccessKey,"XXX");
        // [illegible] SecretKey[illegible]
        properties.put(PropertyKeyConst.SecretKey, "XXX");
        // [illegible] TCP [illegible]
        properties.put(PropertyKeyConst.ONSAddr,
          "http://onsaddr-internet.aliyun.com/rocketmq/nsaddr4client-internet");
        Producer producer = ONSFactory.createProducer(properties);
        // [illegible] start [illegible] Producer[illegible]
        producer.start();
        //[illegible]
        while(true){
            Message msg = new Message( //
                // [illegible] Topic[illegible] Topic [illegible]
                "TopicTestMQ",
                // Message Tag,
                // [illegible] Gmail [illegible] Consumer [illegible]
                   [illegible] MQ [illegible]
                "TagA",
                // Message Body
                // [illegible] MQ [illegible]
                // [illegible] Producer [illegible] Consumer [illegible]
                "Hello MQ".getBytes());
            // [illegible]
               [illegible] MQ [illegible]
            // [illegible]
            msg.setKey("ORDERID_100");
            // [illegible]
            // [illegible] Message ID[illegible]
            SendResult sendResult = producer.send(msg);
            System.out.println("Send Message success. Message ID is: " +
sendResult.getMessageId());
```

```
            }
            // [illegible] Producer [illegible]
            // [illegible]
            producer.shutdown();
        }
    }
```

[illegible]Producer[illegible]GroupName[illegible]NameServer[illegible]Key[illegible]Secret[illegible]MQ[illegible]https://cn.aliyun.com/product/ons[illegible]



[illegible]8-1 [illegible]RocketMQ[illegible]

8.3 RocketMQ□Spark□Flink□□

Spark□Flink□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ RocketMQ□Consumer□Producer□□□□RocketMQ□□□□□□□□□□□□□□□Spark□Flink□Client□□□□□□Spark□Flink□□□

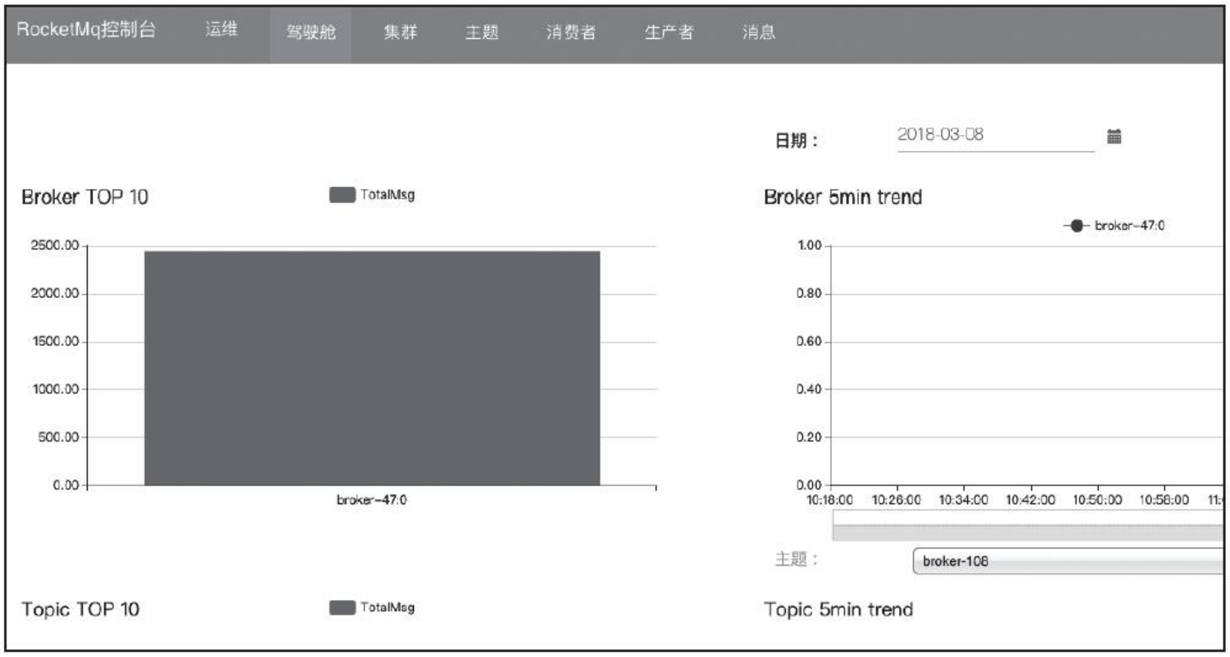
□□□□□□Spark□Flink□□□□□□□□□□□□□□□□Connector□ RocketMQ□Spark□Connector□□□□□□□□ https://github.com/apache/rocketmq-externals/tree/master/rocketmq-spark □RocketMQ□Flink□ Connector□□□□□□□□□□□□□□□□□□□□□□□□

8.4 [illegible]

[illegible]RocketMQ[illegible]

8.4.1 [illegible]

[illegible]1[illegible]7[illegible]Tab[illegible]Topic[illegible]Consumer[illegible]Producer[illegible]8-2[illegible]



[illegible]8-2 RocketMQ[illegible]

[illegible]NaveServer[illegible]VIPChannel[illegible]RocketMQ[illegible]3.5.8[illegible]VIPChannel[illegible]3.5.8[illegible]

[illegible]Broker[illegible]/5[illegible]/[illegible]

[illegible]Broker[illegible]

Topic[illegible]/[illegible]/[illegible]/[illegible]MQAdmin[illegible]updateTopc[illegible]Broker[illegible]Broker[illegible]Message Queue[illegible]Offset[illegible]

Consumer[illegible]/[illegible]/TPS/[illegible]/[illegible]

Producer[illegible]Topic[illegible]Group[illegible]

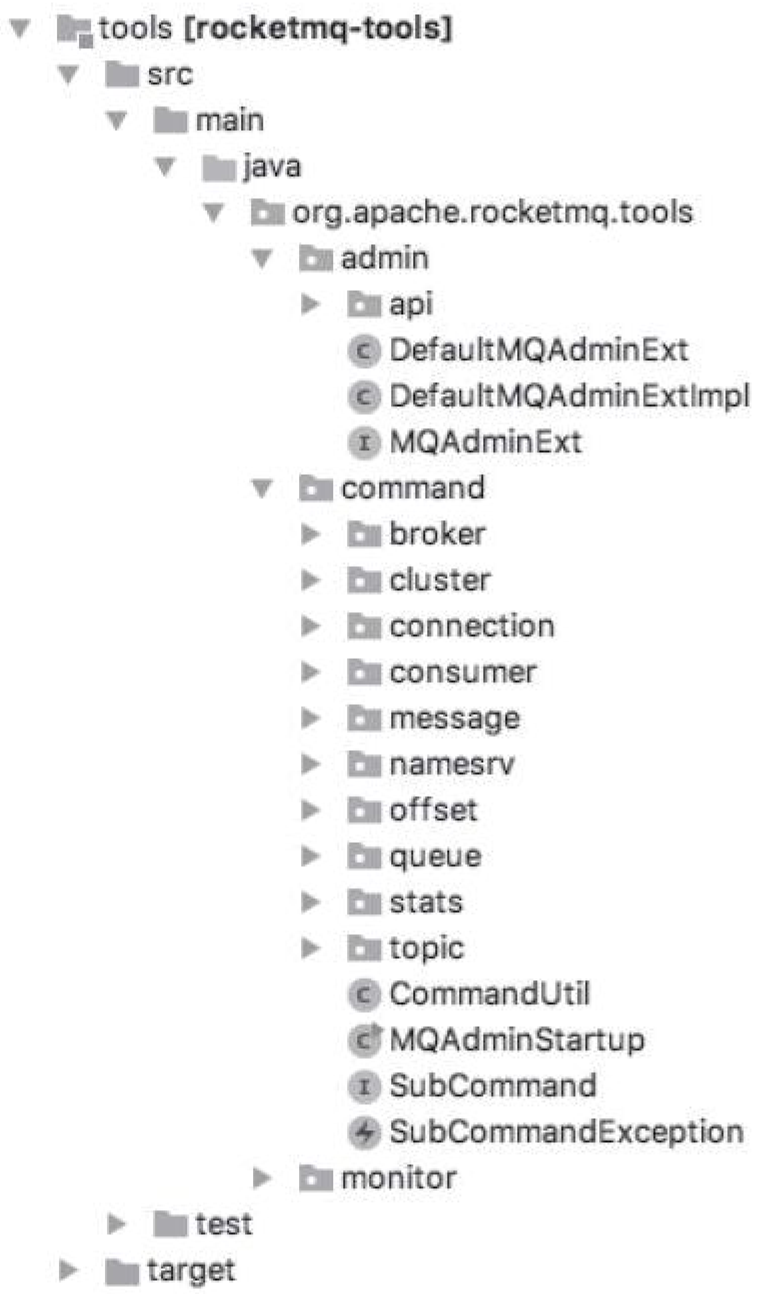
[illegible]Topic[illegible]Key[illegible]ID[illegible]

8.4.2 □□Tools□□□□□□□□□□□

RocketMQ-Console□□□□□Spring Boot□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

RocketMQ□□□□□□Tools□□□MQAdmin□□□□□□□□□□□□□□□□□□□MQAdmin□□□□□□□□□□□□□□□□□□□□□□RocketMQ□Tools□□□□8-3□□□

Tools□□□□□□□□command□□□RocketMQ□□□Java□□□□□□□□Kafka□Scala□□□RabbitMQ□Erlang□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□



```
▼ tools [rocketmq-tools]
  ▼ src
    ▼ main
      ▼ java
        ▼ org.apache.rocketmq.tools
          ▼ admin
            ▶ api
              DefaultMQAdminExt
              DefaultMQAdminExtImpl
              MQAdminExt
          ▼ command
            ▶ broker
            ▶ cluster
            ▶ connection
            ▶ consumer
            ▶ message
            ▶ namesrv
            ▶ offset
            ▶ queue
            ▶ stats
            ▶ topic
              CommandUtil
              MQAdminStartup
              SubCommand
              SubCommandException
          ▶ monitor
    ▶ test
  ▶ target
```

□8-3　RocketMQ□Tools□□

8.5 [illegible]

[illegible] SpringBoot[illegible]Spark[illegible]Flink[illegible]RocketMQ[illegible]RocketMQ[illegible]

□9□　□□Apache□□□□□□□

□□□1□8□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□RocketMQ□□□RocketMQ□□□□□□

9.1 RocketMQ[illegible]

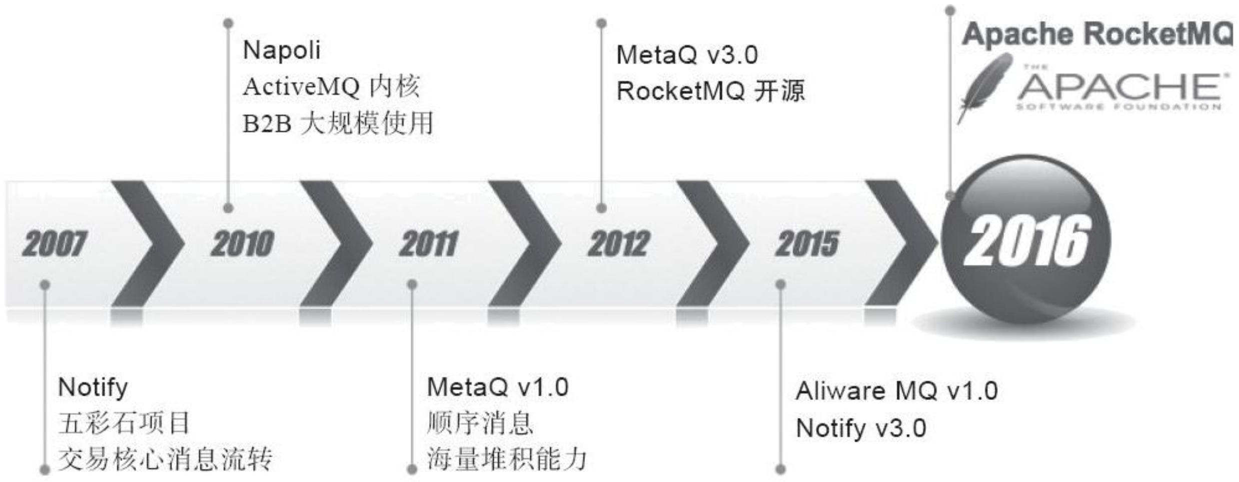
[illegible]2001[illegible]Notify[illegible]

2010[illegible]B2B[illegible]ActiveMQ[illegible]MetaQ 1.0[illegible]2011[illegible]

2012[illegible]MetaQ[illegible]3.0[illegible]RocketMQ[illegible]RocketMQ[illegible]

2015[illegible]RocketMQ[illegible]RocketMQ[illegible]Aliware MQ 1.0[illegible]

2016[illegible]MetaQ[illegible]RocketMQ[illegible]Apache[illegible]



[illegible]9-1 RocketMQ[illegible]

9.2 Apache[illegible]TLP[illegible]

RocketMQ[illegible]Apache Hadoop[illegible]OpenStack[illegible]Redhat[illegible]CentOS[illegible]Fedora[illegible]RocketMQ[illegible]

[illegible]Apache[illegible]Community over Code[illegible]“[illegible]”[illegible]

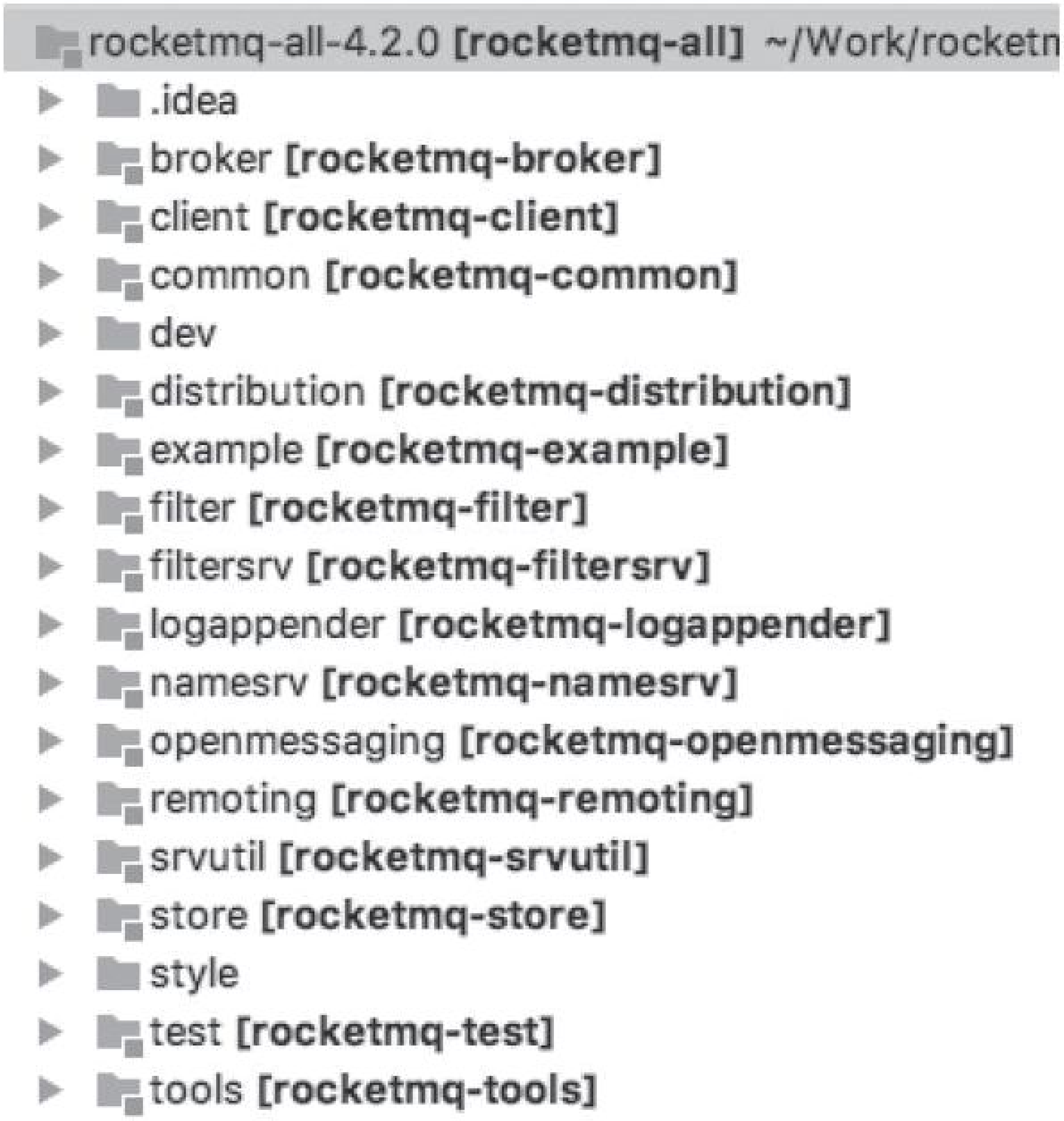
RocketMQ[illegible]Apache[illegible]3[illegible]RocketMQ[illegible]Active Contributors[illegible]Apache[illegible]RocketMQ[illegible]GitHub[illegible]sidebar[illegible]User Guide[illegible]Quick Start[illegible]Architecture & Design[illegible]How to contribute[illegible]Community[illegible]FAQ[illegible]GBK[illegible]UTF-8[illegible]API JavaDoc[illegible]JDepend[illegible]Findbugs[illegible]Release[illegible]New Features[illegible]Improvement[illegible]Bug[illegible]Release note[illegible]

[illegible]RocketMQ[illegible]3.0[illegible]4.0[illegible]4.0[illegible]3.0[illegible]Apache[illegible]Review[illegible]

9.3 □□□□

RocketMQ□□□□□□□9-2□□□□□□□□□Maven□□□□□□□□□□□□□□□□□□□broker□client□common□namesrv□remoting□store□tools□□□□□□□□

namesrv□broker□client□□□□□□□□□□□□namesrv□□□□□□□□□□□□□broker□□□□□□□□□□□client□□□□□□□□□□□□□□□□□□□□□□□□□□common□□□□□□□□□□□□□□□remoting□□□□□□□□□□□store□□□□□□□□□tools□□□□□□□□□□□□□□□



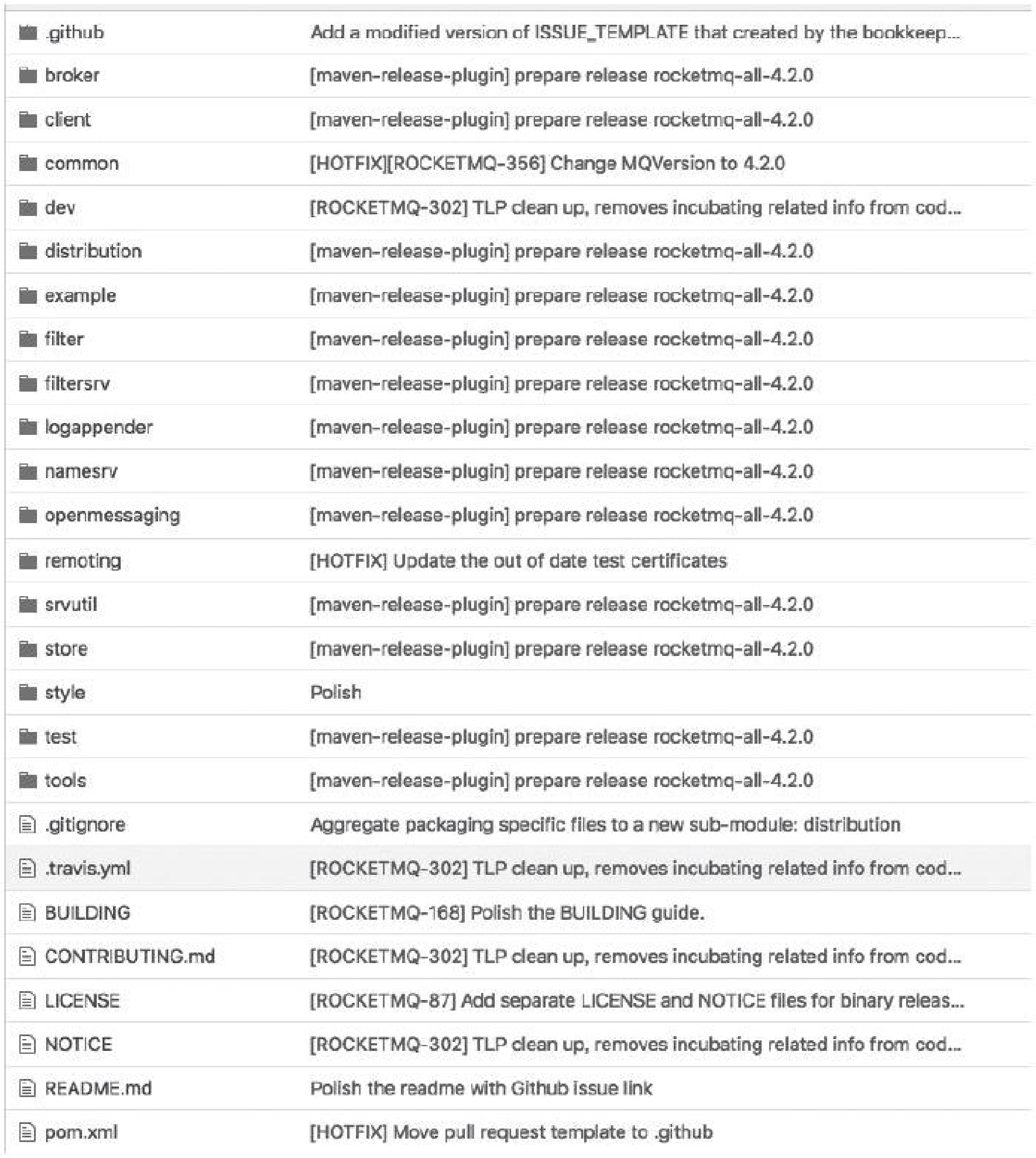
```
rocketmq-all-4.2.0 [rocketmq-all]  ~/Work/rocketn
▶ .idea
▶ broker [rocketmq-broker]
▶ client [rocketmq-client]
▶ common [rocketmq-common]
▶ dev
▶ distribution [rocketmq-distribution]
▶ example [rocketmq-example]
▶ filter [rocketmq-filter]
▶ filtersrv [rocketmq-filtersrv]
▶ logappender [rocketmq-logappender]
▶ namesrv [rocketmq-namesrv]
▶ openmessaging [rocketmq-openmessaging]
▶ remoting [rocketmq-remoting]
▶ srvutil [rocketmq-srvutil]
▶ store [rocketmq-store]
▶ style
▶ test [rocketmq-test]
▶ tools [rocketmq-tools]
```

□9-2　RocketMQ□□□□

9.4 □□□□□□□

RocketMQ□□□□GitHub□□□□□□□□□GitHub□□□□□□□□□□□□
RocketMQ□GitHub□□□□□https://github.com/apache/rocketmq
□GitHub□□□□□□□□□□□□□□□9-3□□

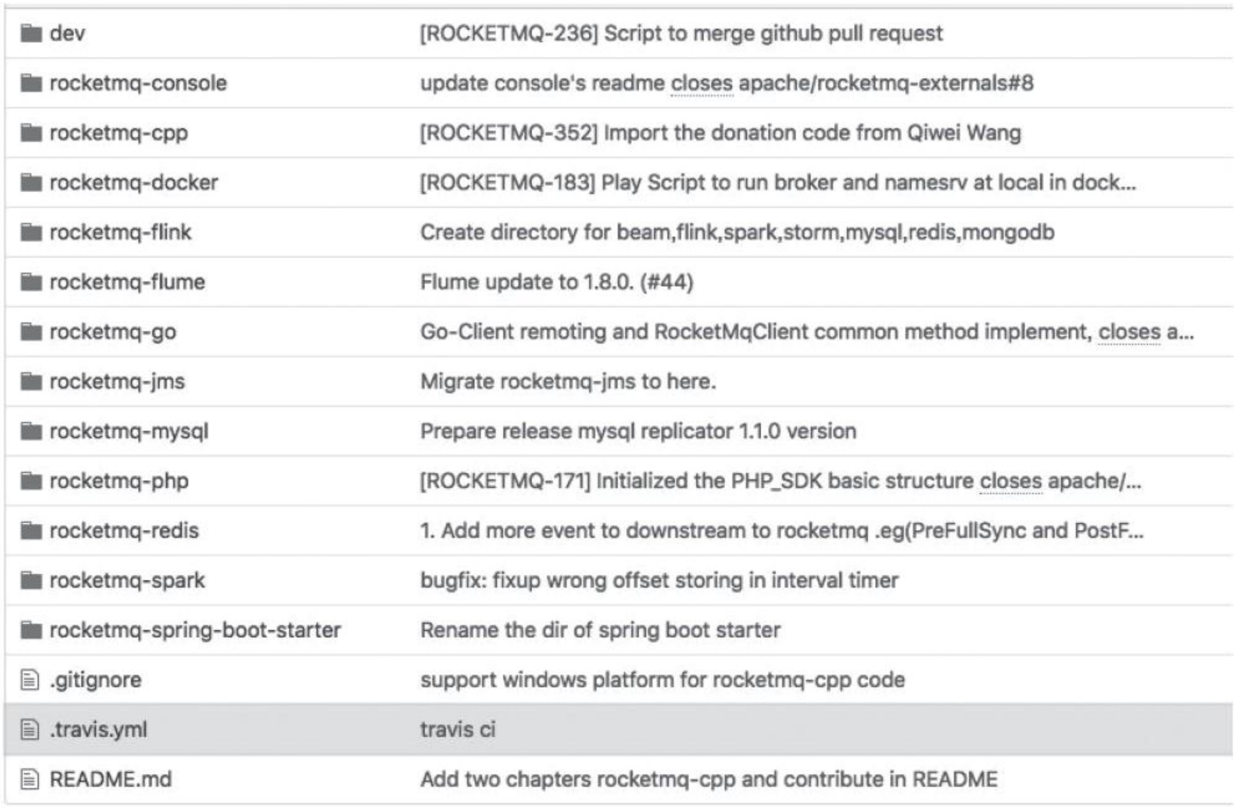
□□RocketMQ□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□
Redis□Spark□Flink□□□□□□□□□□□□□□□□□□□GitHub□□□□□□
https://github.com/apache/rocketmq-externals □



.github	Add a modified version of ISSUE_TEMPLATE that created by the bookkeep...
broker	[maven-release-plugin] prepare release rocketmq-all-4.2.0
client	[maven-release-plugin] prepare release rocketmq-all-4.2.0
common	[HOTFIX][ROCKETMQ-356] Change MQVersion to 4.2.0
dev	[ROCKETMQ-302] TLP clean up, removes incubating related info from cod...
distribution	[maven-release-plugin] prepare release rocketmq-all-4.2.0
example	[maven-release-plugin] prepare release rocketmq-all-4.2.0
filter	[maven-release-plugin] prepare release rocketmq-all-4.2.0
filtersrv	[maven-release-plugin] prepare release rocketmq-all-4.2.0
logappender	[maven-release-plugin] prepare release rocketmq-all-4.2.0
namesrv	[maven-release-plugin] prepare release rocketmq-all-4.2.0
openmessaging	[maven-release-plugin] prepare release rocketmq-all-4.2.0
remoting	[HOTFIX] Update the out of date test certificates
srvutil	[maven-release-plugin] prepare release rocketmq-all-4.2.0
store	[maven-release-plugin] prepare release rocketmq-all-4.2.0
style	Polish
test	[maven-release-plugin] prepare release rocketmq-all-4.2.0
tools	[maven-release-plugin] prepare release rocketmq-all-4.2.0
.gitignore	Aggregate packaging specific files to a new sub-module: distribution
.travis.yml	[ROCKETMQ-302] TLP clean up, removes incubating related info from cod...
BUILDING	[ROCKETMQ-168] Polish the BUILDING guide.
CONTRIBUTING.md	[ROCKETMQ-302] TLP clean up, removes incubating related info from cod...
LICENSE	[ROCKETMQ-87] Add separate LICENSE and NOTICE files for binary releas...
NOTICE	[ROCKETMQ-302] TLP clean up, removes incubating related info from cod...
README.md	Polish the readme with Github issue link
pom.xml	[HOTFIX] Move pull request template to .github

□9-3 RocketMQ□GitHub□□□□□

□□□□□□□□□□□□□□□□□□□□□ http://rocketmq.apache.org/docs/how-to-contribute/ □□□□□□□□□□□PR□□□□□□□□□□□□□□□□□□PR□□□□□□□□□□□□□□□□□□□□



dev	[ROCKETMQ-236] Script to merge github pull request
rocketmq-console	update console's readme closes apache/rocketmq-externals#8
rocketmq-cpp	[ROCKETMQ-352] Import the donation code from Qiwei Wang
rocketmq-docker	[ROCKETMQ-183] Play Script to run broker and namesrv at local in dock...
rocketmq-flink	Create directory for beam,flink,spark,storm,mysql,redis,mongodb
rocketmq-flume	Flume update to 1.8.0. (#44)
rocketmq-go	Go-Client remoting and RocketMqClient common method implement, closes a...
rocketmq-jms	Migrate rocketmq-jms to here.
rocketmq-mysql	Prepare release mysql replicator 1.1.0 version
rocketmq-php	[ROCKETMQ-171] Initialized the PHP_SDK basic structure closes apache/...
rocketmq-redis	1. Add more event to downstream to rocketmq .eg(PreFullSync and PostF...
rocketmq-spark	bugfix: fixup wrong offset storing in interval timer
rocketmq-spring-boot-starter	Rename the dir of spring boot starter
.gitignore	support windows platform for rocketmq-cpp code
.travis.yml	travis ci
README.md	Add two chapters rocketmq-cpp and contribute in README

□9-4 rocket-externals□□□□

9.5 [illegible]

RocketMQ[illegible]RocketMQ[illegible]Apache[illegible]NameServer[illegible]

[illegible]10[illegible] NameServer[illegible]

[illegible]4[illegible]NameServer[illegible]NameServer[illegible]NameServer[illegible]

10.1 [illegible]

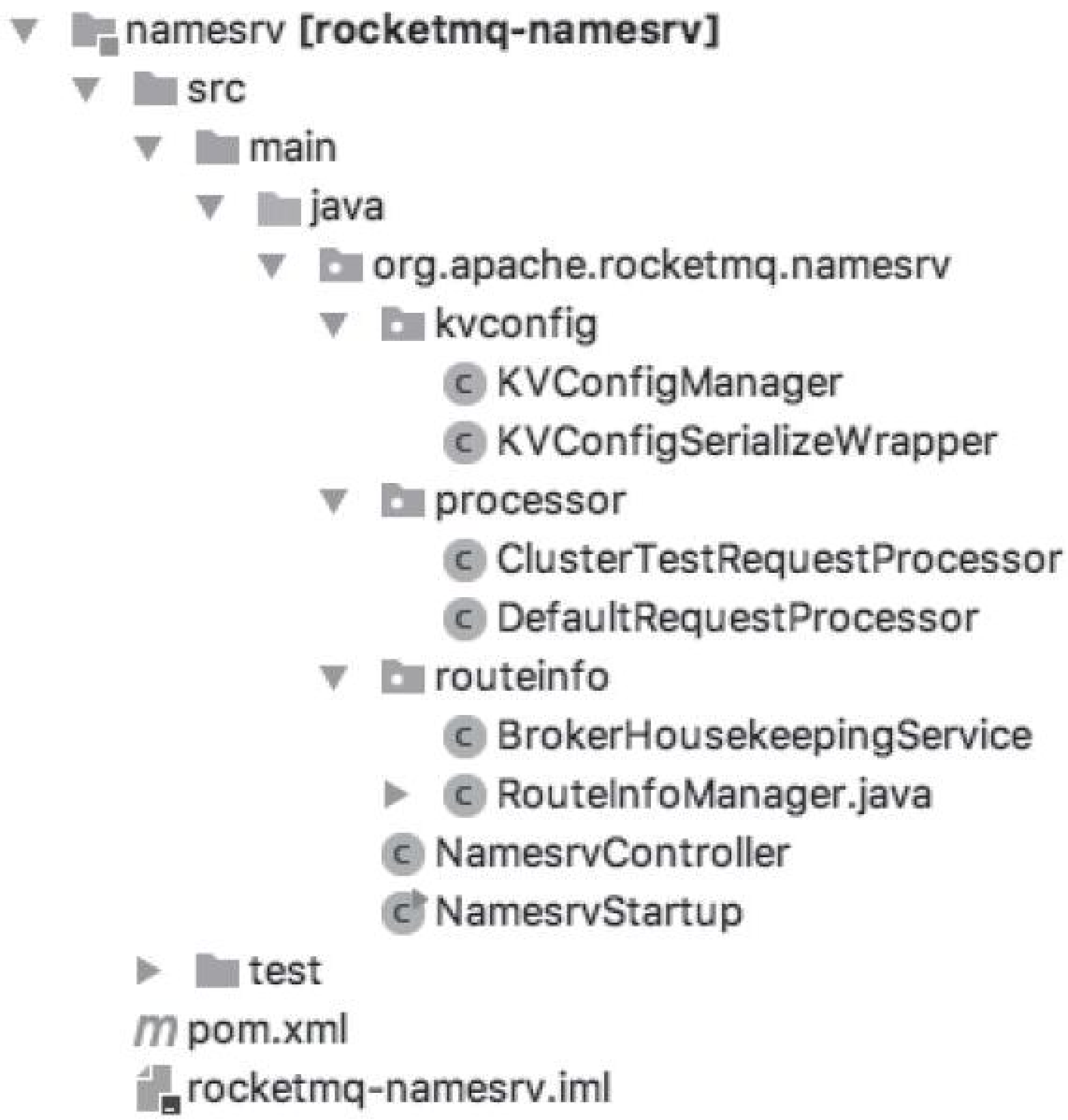
[illegible]NameServer[illegible]Controller[illegible]

10.1.1 □□□□

□□□□□NameServer□□□□□□□□10-1□□

NamesrvStartup□□□□□□□□□NamesrvController□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□NamesrvStartup.java□□main□□public static void main□String[]args□{main0□args□□}□□□□□□□□□□main0□□□□□□



```
▼ namesrv [rocketmq-namesrv]
  ▼ src
    ▼ main
      ▼ java
        ▼ org.apache.rocketmq.namesrv
          ▼ kvconfig
              KVConfigManager
              KVConfigSerializeWrapper
          ▼ processor
              ClusterTestRequestProcessor
              DefaultRequestProcessor
          ▼ routeinfo
              BrokerHousekeepingService
            ▶ RouteInfoManager.java
            NamesrvController
            NamesrvStartup
    ▶ test
    pom.xml
    rocketmq-namesrv.iml
```

□10-1 NameServer□□□□

10.1.2 [illegible]

main0[illegible]-c[illegible]-p[illegible]10-1[illegible]

[illegible]10-1 [illegible]NameServer[illegible]

```
Options options = ServerUtil.buildCommandlineOptions(new Options());
commandLine = ServerUtil.parseCmdLine("mqnamesrv", args,
    buildCommandlineOptions(options), new PosixParser());
if (null == commandLine) {
    System.exit(-1);
    return null;
}
final NamesrvConfig namesrvConfig = new NamesrvConfig();
final NettyServerConfig nettyServerConfig = new NettyServerConfig();
nettyServerConfig.setListenPort(9876);
if (commandLine.hasOption('c')) {
    String file = commandLine.getOptionValue('c');
    if (file != null) {
        InputStream in = new BufferedInputStream(new
            FileInputStream(file));
        properties = new Properties();
        properties.load(in);
        MixAll.properties2Object(properties, namesrvConfig);
        MixAll.properties2Object(properties, nettyServerConfig);
        namesrvConfig.setConfigStorePath(file);
        System.out.printf("load config properties file OK, " +
            file + "%n");
        in.close();
    }
}
if (commandLine.hasOption('p')) {
    MixAll.printObjectProperties(null, namesrvConfig);
    MixAll.printObjectProperties(null, nettyServerConfig);
    System.exit(0);
}
```

-c[illegible]-p[illegible]-p[illegible]

10.1.3 □□□NameServer□Controller

main0□□□□□□□□□□□□□Controller□□□□□□10-2□□□

□□□□10-2 □□□□□□Controller

```
// remember all configs to prevent discard
controller.getConfiguration().registerConfig(properties);
boolean initResult = controller.initialize();
if (!initResult) {
    controller.shutdown();
    System.exit(-3);
}
Runtime.getRuntime().addShutdownHook(new ShutdownHookThread(log,
    new Callable<Void>() {
    @Override
    public Void call() throws Exception {
        controller.shutdown();
        return null;
    }
}));
controller.start();
```

□□□□□□□□□□□□□controller.initialize□□□□□□□□□□□
controller.start□□□NameServer□□□□□

□□□□□□□□□ShutdownHookThread□□□□□□□□□□□□
controller.shutdown□□□□□□□□□□□

10.2 NameServer□□□□□

NameServer□□□□□□NamesrvController.java□□□□NameServer□□NameserverController□□□□□□□□□□□□□□□□10-3□□□

□□□□10-3 □□□□□□

```
this.remotingExecutor =
    Executors.newFixedThreadPool(nettyServerConfig
        .getServerWorkerThreads(), new ThreadFactoryImpl
        ("RemotingExecutorThread_"));
this.registerProcessor();

this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
    @Override
    public void run() {
       NamesrvController.this.routeInfoManager.scanNotActiveBroker();
    }
}, 5, 10, TimeUnit.SECONDS);
this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
    @Override
    public void run() {
       NamesrvController.this.kvConfigManager.printAllPeriodically();
    }
}, 1, 10, TimeUnit.MINUTES);
```

□□□□□□□□8□□□□□□□□private int serverWorkerThreads=8□□□□□□□□□□□□□□□□□□□□□□□Broker□scanNotActiveBroker□□□□□□□□□□□□□□printAllPeriodically□□

□□□□□□□□□□□remotingServer□remotingServer□□□□□□□□□Broker□Client□□□□□□□□□□□□□□□□□□□□□□Processor□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□10-4□□□

□□□□10-4 □□□□□□□□□□□□□□□□

```
this.remotingServer = new NettyRemotingServer(this.nettyServerConfig,
    this.brokerHousekeepingService);
……
if (namesrvConfig.isClusterTest()) {
    this.remotingServer.registerDefaultProcessor(new
            ClusterTestRequestProcessor(this, namesrvConfig
            .getProductEnvName()),
        this.remotingExecutor);
} else {
    this.remotingServer.registerDefaultProcessor(new
        DefaultRequestProcessor(this), this.remotingExecutor);
}
```

remotingServer□□□Netty□□□□□□□□□□□□□□□remoting-Server□□□□Netty□□□□□□□□□□□□□□□□

10.3 [illegible]

NameServer[illegible]DefaultRequestProcessor.java[illegible]Processor[illegible]10-5[illegible]

[illegible]10-5 [illegible]

```
switch (request.getCode()) {
    case RequestCode.PUT_KV_CONFIG:
        return this.putKVConfig(ctx, request);
    case RequestCode.GET_KV_CONFIG:
        return this.getKVConfig(ctx, request);
    case RequestCode.DELETE_KV_CONFIG:
        return this.deleteKVConfig(ctx, request);
    case RequestCode.REGISTER_BROKER:
        Version brokerVersion = MQVersion.value2Version(request
            .getVersion());
        if (brokerVersion.ordinal() >= MQVersion.Version
            .V3_0_11.ordinal()) {
            return this.registerBrokerWithFilterServer(ctx, request);
        } else {
            return this.registerBroker(ctx, request);
        }
    case RequestCode.UNREGISTER_BROKER:
        return this.unregisterBroker(ctx, request);
    case RequestCode.GET_ROUTEINTO_BY_TOPIC:
        return this.getRouteInfoByTopic(ctx, request);
    case RequestCode.GET_BROKER_CLUSTER_INFO:
        return this.getBrokerClusterInfo(ctx, request);
    case RequestCode.WIPE_WRITE_PERM_OF_BROKER:
        return this.wipeWritePermOfBroker(ctx, request);
    case RequestCode.GET_ALL_TOPIC_LIST_FROM_NAMESERVER:
        return getAllTopicListFromNameserver(ctx, request);
    case RequestCode.DELETE_TOPIC_IN_NAMESRV:
        return deleteTopicInNamesrv(ctx, request);
    case RequestCode.GET_KVLIST_BY_NAMESPACE:
        return this.getKVListByNamespace(ctx, request);
    case RequestCode.GET_TOPICS_BY_CLUSTER:
        return this.getTopicsByCluster(ctx, request);
    case RequestCode.GET_SYSTEM_TOPIC_LIST_FROM_NS:
        return this.getSystemTopicListFromNs(ctx, request);
    case RequestCode.GET_UNIT_TOPIC_LIST:
        return this.getUnitTopicList(ctx, request);
    case RequestCode.GET_HAS_UNIT_SUB_TOPIC_LIST:
        return this.getHasUnitSubTopicList(ctx, request);
    case RequestCode.GET_HAS_UNIT_SUB_UNUNIT_TOPIC_LIST:
        return this.getHasUnitSubUnUnitTopicList(ctx, request);
    case RequestCode.UPDATE_NAMESRV_CONFIG:
        return this.updateConfig(ctx, request);
    case RequestCode.GET_NAMESRV_CONFIG:
        return this.getConfig(ctx, request);
    default:
        break;
}
```

[illegible]switch[illegible]RequestCode[illegible]RequestCode[illegible]NameServer[illegible]REGISTER_BROKER[illegible]Broker[illegible]GET_ROUTEINTO_BY_TOPIC[illegible]Topic[illegible]WIPE_WRITE_PERM_OF_BROKER[illegible]Broker[illegible]

10.4 □□□□□

NameServer□□□□□□□□□□□□□□□□□□
RouteInfoManager□□□□□□□10-6□□

□□□□10-6 RouteInfoManager□□□□□

```
private final HashMap<String/* topic */, List<QueueData>> topicQueue-Table;
private final HashMap<String/* brokerName */, BrokerData> brokerAddr-Table;
private final HashMap<String/* clusterName */, Set<String/* brokerName
*/>> clusterAddrTable;
private final HashMap<String/* brokerAddr */, BrokerLiveInfo>
    brokerLiveTable;
private final HashMap<String/* brokerAddr */, List<String>/* Filter
Server */> filterServerTable;
public RouteInfoManager() {
    this.topicQueueTable = new HashMap<String, List<QueueData>>(1024);
    this.brokerAddrTable = new HashMap<String, BrokerData>(128);
    this.clusterAddrTable = new HashMap<String, Set<String>>(32);
    this.brokerLiveTable = new HashMap<String, BrokerLiveInfo>(256);
    this.filterServerTable = new HashMap<String, List<String>>(256);
}
```

□□□□□□□□□□□□5□□□□RocketMQ□□□□□□□
□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□NameServer□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□lock□□□□□□□□□□□□□10-7□□

□□□□10-7 □□□□□□□

```
Lock lock = new Lock();
public void outer() {
    lock.lock();
    inner();
    lock.unlock();
}
public void inner() {
    lock.lock();
    //do something lock.unlock(); }
}
```

RouteInfoManager□□□□□□□□□□□□□private final ReadWriteLock lock=new ReentrantReadWriteLock□□□□□□□deleteTopic□□□□□□□□□□□□□□□□□□□□10-8□□□

□□□□10-8　□□□□□□

```
public void deleteTopic(final String topic) {
    try {
        try {
            this.lock.writeLock().lockInterruptibly();
            this.topicQueueTable.remove(topic);
        } finally {
            this.lock.writeLock().unlock();
        }
    } catch (Exception e) {
        log.error("deleteTopic Exception", e);
    }
}
```

□□□□□□□□□□□□□□□□try{}□□□□□finally{}□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

10.5 [illegible]

[illegible]NameServer[illegible]NameServer[illegible]Client[illegible]RocketMQ[illegible]Client[illegible]

□11□　□□□□□□□

□□□□□□RocketMQ□□□□□□□□□□□□□DefaultMQPush-Consumer□□

11.1 □□□□

□□□□DefaultMQPushConsumer□□□□□□□□□□□□GroupName□NameServer□□□□□□□□Topic□□□□□□□Message□□□□□□□□□start□□□□□□□□□□□□□□□□□

11.1.1 □□□□□

DefaultMQPushConsumer□□ org.apache.rocketmq.client.consumer□□□□□□□□□□□□□□□□□□□□□□□DefaultMQPushConsumerImpl□□□□□□□□11-1□□□

□□□□11-1 DefaultMQPushConsumer□□□

```
/**
 * Default constructor.
 */
public DefaultMQPushConsumer() {
    this(MixAll.DEFAULT_CONSUMER_GROUP, null, new
        AllocateMessageQueueAveragely());
}
/**
 * Constructor specifying consumer group, RPC hook and message queue
 * allocating algorithm.
 *
 * @param consumerGroup Consume queue.
 * @param rpcHook RPC hook to execute before each remoting command.
 * @param allocateMessageQueueStrategy message queue allocating algorithm.
 */
public DefaultMQPushConsumer(final String consumerGroup, RPCHook rpcHook,
    AllocateMessageQueueStrategy allocateMessageQueueStrategy) {
    this.consumerGroup = consumerGroup;
    this.allocateMessageQueueStrategy = allocateMessageQueueStrategy;
    defaultMQPushConsumerImpl = new DefaultMQPushConsumerImpl(this,
        rpcHook);
}
/**
 * Constructor specifying RPC hook.
 *
 * @param rpcHook RPC hook to execute before each remoting command.
 */
public DefaultMQPushConsumer(RPCHook rpcHook) {
    this(MixAll.DEFAULT_CONSUMER_GROUP, rpcHook, new
        AllocateMessageQueueAveragely());
}

/**
 * Constructor specifying consumer group.
 *
 * @param consumerGroup Consumer group.
 */
public DefaultMQPushConsumer(final String consumerGroup) {
    this(consumerGroup, null, new AllocateMessageQueueAveragely());
}
```

□□□□□□□□□□□□□□□□□□□□consumer Group□□□□□□□□□□□□□□□RPCHoop□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

DefaultMQPushConsumerImpl[illegible]

11.1.2 DefaultMQPushConsumer□□□□

DefaultMQPushConsumerImpl□□□□□ DefaultMQPushConsumer□□□□□□ DefaultMQPushConsumerImpl.java□ org.apache.rocketmq.client.impl.consumer□□□□□□□□□□□start □□□□□□□

□□□□□□MQClientInstance□□□□□□rebalance□□□pullApi-Wraper□□□□□□□□□□□pull□□□□□□□□□□□□11-2□□□

□□□□11-2 □□□MQClientInstance□pullApiWraper

```
this.mQClientFactory = MQClientManager.getInstance()
    .getAndCreateMQClientInstance(this.defaultMQPushConsumer,
        this.rpcHook);
this.rebalanceImpl.setConsumerGroup(this
    .defaultMQPushConsumer.getConsumerGroup());
this.rebalanceImpl.setMessageModel(this.defaultMQPushConsumer
    .getMessageModel());
this.rebalanceImpl.setAllocateMessageQueueStrategy(this
    .defaultMQPushConsumer.getAllocateMessageQueueStrategy());
this.rebalanceImpl.setmQClientFactory(this.mQClientFactory);
this.pullAPIWrapper = new PullAPIWrapper(
    mQClientFactory,
    this.defaultMQPushConsumer.getConsumerGroup(), isUnitMode
    ());
this.pullAPIWrapper.registerFilterMessageHook
    (filterMessageHookList);
```

□□□□□OffsetStore□OffsetStore□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□11-3□□□

□□□□11-3 □□□OffsetStore

```
if (this.defaultMQPushConsumer.getOffsetStore() != null) {
    this.offsetStore = this.defaultMQPushConsumer
        .getOffsetStore();
} else {
    switch (this.defaultMQPushConsumer.getMessageModel()) {
        case BROADCASTING:
            this.offsetStore = new LocalFileOffsetStore(this
                .mQClientFactory, this.defaultMQPushConsumer
                .getConsumerGroup());
            break;
        case CLUSTERING:
            this.offsetStore = new RemoteBrokerOffsetStore
                (this.mQClientFactory, this
                    .defaultMQPushConsumer.getConsumerGroup());
            break;
        default:
            break;
    }
    this.defaultMQPushConsumer.setOffsetStore(this.offsetStore);
}
this.offsetStore.load();
```

□□□□□□□□□□□□OffsetStore□□□□□□□□□□BROADCASTING□□□□□□□LocalFileOffsetStore□Offset□□□□□□□□CLUSTERING□□□□□□□RemoteBrokerOffsetStore□Offset□□Broker□□□□

□□□□□□consumeMessageService□□□□□□□□□□□□□□□□□□□Service□□□□□□□□11-4□□□

□□□□11-4　□□□consumeMessageService

```
    if (this.getMessageListenerInner() instanceof
        MessageListenerOrderly) {
        this.consumeOrderly = true;
        this.consumeMessageService =
            new ConsumeMessageOrderlyService(this,
                (MessageListenerOrderly) this
                    .getMessageListenerInner());
    } else if (this.getMessageListenerInner() instanceof
        MessageListenerConcurrently) {
        this.consumeOrderly = false;
        this.consumeMessageService =
            new ConsumeMessageConcurrentlyService(this,
                (MessageListenerConcurrently) this
                    .getMessageListenerInner());
    }
    this.consumeMessageService.start();
```

□□□□MQClientInstance□start□□□□□□□□□□

11.1.3 □□□□□□

□□□□□□□□□□public void pullMessage□final PullRequest pullRequest□□□11-5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□11-5 □□□□□□

```
if (cachedMessageCount > this.defaultMQPushConsumer
    .getPullThresholdForQueue()) {
    this.executePullRequestLater(pullRequest,
        PULL_TIME_DELAY_MILLS_WHEN_FLOW_CONTROL);
    if ((queueFlowControlTimes++ % 1000) == 0) {
        log.warn(
            "the cached message count exceeds the threshold {}, so do" +
                " flow control, minOffset={}, maxOffset={}, count={}," +
                " size={} MiB, pullRequest={}, flowControlTimes={}",
            this.defaultMQPushConsumer.getPullThresholdForQueue(),
            processQueue.getMsgTreeMap().firstKey(), processQueue
                .getMsgTreeMap().lastKey(), cachedMessageCount,
            cachedMessageSizeInMiB, pullRequest, queueFlowControlTimes);
    }
    return;
}
if (cachedMessageSizeInMiB > this.defaultMQPushConsumer
    .getPullThresholdSizeForQueue()) {
    this.executePullRequestLater(pullRequest,
        PULL_TIME_DELAY_MILLS_WHEN_FLOW_CONTROL);
    if ((queueFlowControlTimes++ % 1000) == 0) {
        log.warn(
            "the cached message size exceeds the threshold {} MiB, so" +
                " do flow control, minOffset={}, maxOffset={}, " +
                "count={}, size={} MiB, pullRequest={}, " +
                "flowControlTimes={}",
            this.defaultMQPushConsumer.getPullThresholdSizeForQueue()
            , processQueue.getMsgTreeMap().firstKey(), processQueue
                .getMsgTreeMap().lastKey(), cachedMessageCount,
            cachedMessageSizeInMiB, pullRequest, queueFlowControlTimes);
    }
    return;
}
```

□□□11-6□□□

□□□□11-6 □□□□□□□□□□

```
switch (pullResult.getPullStatus()) {
    case FOUND:
        long prevRequestOffset = pullRequest
            .getNextOffset();
        pullRequest.setNextOffset(pullResult
            .getNextBeginOffset());
        ……
        break;
    case NO_NEW_MSG:
        pullRequest.setNextOffset(pullResult
            .getNextBeginOffset());
        DefaultMQPushConsumerImpl.this.correctTagsOffset
            (pullRequest);
        DefaultMQPushConsumerImpl.this
            .executePullRequestImmediately(pullRequest);
        break;
    case NO_MATCHED_MSG:
        pullRequest.setNextOffset(pullResult
            .getNextBeginOffset());
        DefaultMQPushConsumerImpl.this.correctTagsOffset
            (pullRequest);
        DefaultMQPushConsumerImpl.this
            .executePullRequestImmediately(pullRequest);
        break;
    case OFFSET_ILLEGAL:
        log.warn("the pull request offset illegal, {} {}",
            pullRequest.toString(), pullResult.toString());
        pullRequest.setNextOffset(pullResult
            .getNextBeginOffset());
……        break;
    default:
        break;
}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□11-7□□□

□□□□11-7   □□pull□□

```
try {
    this.pullAPIWrapper.pullKernelImpl(
        pullRequest.getMessageQueue(),
        subExpression,
        subscriptionData.getExpressionType(),
        subscriptionData.getSubVersion(),
        pullRequest.getNextOffset(),
        this.defaultMQPushConsumer.getPullBatchSize(),
        sysFlag,
        commitOffsetValue,
        BROKER_SUSPEND_MAX_TIME_MILLIS,
        CONSUMER_TIMEOUT_MILLIS_WHEN_SUSPEND,
        CommunicationMode.ASYNC,
        pullCallback
    );
} catch (Exception e) {
    log.error("pullKernelImpl exception", e);
    this.executePullRequestLater(pullRequest,
        PULL_TIME_DELAY_MILLS_WHEN_EXCEPTION);
}
```

11.2 [illegible]

[illegible]

11.2.1 □□□□□

□□□□□□□□□□Consumer□□□□□□□□□□□□□Consume-MessageConcurrentlyService□□□□□□RocketMQ□□□□□□Consume-MessageConcurrentlyService□□org.apache.rocketmq.client.impl.consumer□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□consumeThreadMin□consumeThreadMax□□□15□□□□□□□□□11-8□□□

□□□□11-8 □□□□□

```
this.consumeExecutor = new ThreadPoolExecutor(
    this.defaultMQPushConsumer.getConsumeThreadMin(),
    this.defaultMQPushConsumer.getConsumeThreadMax(), 1000 * 60,
    TimeUnit.MILLISECONDS, this.consumeRequestQueue,
    new ThreadFactoryImpl("ConsumeMessageThread_"));
this.scheduledExecutorService =
    Executors.newSingleThreadScheduledExecutor(new ThreadFactoryImpl(
        "ConsumeMessageScheduledThread_"));
this.cleanExpireMsgExecutors =
    Executors.newSingleThreadScheduledExecutor(new ThreadFactoryImpl(
        "CleanExpireMsgScheduledThread_"));
```

□Broker□□□□□□□□□□□□BatchSize□□□□□□□□□□□□□□ConsumeRequest□□□□□□□ConsumeRequest□□□consumeExecutor□□□□□□□□□□□□11-9□□□

□□□□11-9 □□□□□□

```
if (msgs.size() <= consumeBatchSize) {
    ConsumeRequest consumeRequest = new ConsumeRequest(msgs,
        processQueue, messageQueue);
    try {
        this.consumeExecutor.submit(consumeRequest);
    } catch (RejectedExecutionException e) {
        this.submitConsumeRequestLater(consumeRequest);
    }
} else {
    for (int total = 0; total < msgs.size(); ) {
        List<MessageExt> msgThis = new ArrayList<MessageExt>
```

```
            (consumeBatchSize);
        for (int i = 0; i < consumeBatchSize; i++, total++) {
            if (total < msgs.size()) {
                msgThis.add(msgs.get(total));
            } else {
                break;
            }
        }
        ConsumeRequest consumeRequest = new ConsumeRequest(msgThis,
            processQueue, messageQueue);
        try {
            this.consumeExecutor.submit(consumeRequest);
        } catch (RejectedExecutionException e) {
            for (; total < msgs.size(); total++) {
                msgThis.add(msgs.get(total));
            }

            this.submitConsumeRequestLater(consumeRequest);
        }
    }
}
```

□□□□□□□□□□□□□□□□□□□□□CONSUME_SUCCESS□RECONSUME_LATER□□□□□□□□□□□□□□□□□□□□scheduledExecutorService□□□□□5□□□□□□□□□□□□□CLUSTERING□□□□□□□□□□□□□□□□□Broker□□□□ConsumerGroup□□□□Consumer□□□□□□□□Broker□□□□□□RECONSUME_LATER□□□□□□Status□□□□□□□□□11-10□□□

□□□□11-10　□□□□□Status□□□□

```
switch (this.defaultMQPushConsumer.getMessageModel()) {
    case BROADCASTING:
        for (int i = ackIndex + 1; i < consumeRequest.getMsgs().size
            (); i++) {
            MessageExt msg = consumeRequest.getMsgs().get(i);
            log.warn("BROADCASTING, the message consume failed, drop " +
                "it, {}", msg.toString());
        }
        break;
    case CLUSTERING:
        List<MessageExt> msgBackFailed = new ArrayList<MessageExt>
            (consumeRequest.getMsgs().size());
        for (int i = ackIndex + 1; i < consumeRequest.getMsgs().size
            (); i++) {
            MessageExt msg = consumeRequest.getMsgs().get(i);
            boolean result = this.sendMessageBack(msg, context);
            if (!result) {
                msg.setReconsumeTimes(msg.getReconsumeTimes() + 1);
                msgBackFailed.add(msg);
            }
        }
        if (!msgBackFailed.isEmpty()) {
            consumeRequest.getMsgs().removeAll(msgBackFailed);
            this.submitConsumeRequestLater(msgBackFailed,
                consumeRequest.getProcessQueue(), consumeRequest
```

```
                    .getMessageQueue());
        }
        break;
    default:
        break;
}
```

[illegible]

11.2.2 ProcessQueue□□

□□□□□□□□□□ProcessQueue□□□□□□□□□□□□□□□□□□□Broker□□RocketMQ□□□□□□□□ProcessQueue□□□□□□□□□PushConsumer□□□□□□□□Message Queue□□□□□□□□ProcessQueue□□□□□□□□Message Queue□□□□□□□□□□□□□□□11-11□□□

ProcessQueue□□□□□□□□□□□TreeMap□□□□□□□TreeMap□□Message Queue□Offset□□Key□□□□□□□□□□Value□□□□□□□MessageQueue□□□□□□□□□□□□□□□□□□□□□□□□□TreeMap□□□□□□□□

□□□□11-11 □□□□□□□□□

```
private final ReadWriteLock lockTreeMap = new ReentrantReadWriteLock();
private final TreeMap<Long, MessageExt> msgTreeMap = new TreeMap<Long,
MessageExt>();
private final AtomicLong msgCount = new AtomicLong();
private final AtomicLong msgSize = new AtomicLong();
private final Lock lockConsume = new ReentrantLock();
```

□□ProcessQueue□□□ConsumeMessageOrderlyService□□□□□□□□□□ConsumeMessageConcurrentlyService□□□□□□□□□□□□□□□□□□

11.3 [illegible]

[illegible]Broker[illegible]Broker[illegible]

11.3.1 MQClientInstance[illegible]

MQClientInstance[illegible]Consumer[illegible]Producer[illegible]NameServer[illegible]Topic[illegible]Route[illegible]MQClientInstance[illegible]MQClientAPIImpl[illegible]Broker[illegible]Broker[illegible]

[illegible]MQClientInstance[illegible]Consumer[illegible]Producer[illegible]MQClientInstance[illegible]Consumer[illegible]Producer[illegible]RocketMQ[illegible]MQClientInstance[illegible]MQClientInstance[illegible]11-12[illegible]

[illegible]11-12 [illegible]MQClientInstance

```
MQClientManager.getInstance().getAndCreateMQClientInstance(this.defaultMQProducer
, rpcHook);
```

[illegible]MQClientInstance[illegible]MQClientInstance[illegible]11-13[illegible]

[illegible]11-13 MQClientInstance[illegible]

```
public MQClientInstance getAndCreateMQClientInstance(
    final ClientConfig clientConfig, RPCHook rpcHook) {
    String clientId = clientConfig.buildMQClientId();
    MQClientInstance instance = this.factoryTable.get(clientId);
    if (null == instance) {
        instance =
            new MQClientInstance(clientConfig.cloneClientConfig(),
                this.factoryIndexGenerator.getAndIncrement(), clientId,
                rpcHook);
        MQClientInstance prev = this.factoryTable.putIfAbsent(clientId,
            instance);
        if (prev != null) {
            instance = prev;
            log.warn("Returned Previous MQClientInstance for " +
                "clientId:[{}]", clientId);
        } else {
            log.info("Created new MQClientInstance for clientId:[{}]",
                clientId);
        }
    }
    return instance;
}
```

□□□□□□ConcurrentMap<String/*clientId*/□ MQClientInstance>factoryTable□□Map□□□□□□□□□□MQClientInstance□□□□clientId□□Key□□Map□□□□clientId□□□□“clientIp”+@+“InstanceName”□□□clientIp□□□□□□□IP□□□□□□□□instancename□□□□□□□□□□□□□□

□□□□□□□□□□RocketMQ□□□□Java□□□□□□□□JVM□□□□□□□MQClientInstance□□□□□□□□□□□□□□□□Consumer□□Producer□□□□□□□□□MQClientInstance□□□

□quick start□□□□□□□DefaultMQPushConsumer□□□□□□□□□□□□Consumer□InstanceName□□□□□setInstanceName□□□□□□□□□□□□InstanceName□□□□□□“DEFAULT”□□□□□□MQClientInstance□□□□□□□□□□□□□InstanceName□□□□□□□□□□11-14□□□

□□□□11-14　InstanceName□□□□

```
if (this.defaultMQPushConsumer.getMessageModel() == MessageModel.CLUSTERING) {
    this.defaultMQPushConsumer.changeInstanceNameToPID();
}
public void changeInstanceNameToPID() {
    if (this.instanceName.equals("DEFAULT")) {
        this.instanceName = String.valueOf(UtilAll.getPid());
    }
}
```

□InstanceName□□□□□□□□□□□□□□□Consumer□□Producer□□□□□□□□□□InstanceName□□□□□□□□□MQClientInstance□□□

□□□□□□□□□MQClientInstance□□□□□□□□□□□Java□□□□□□□□RoceketMQ□□□□□□□□□□□□□□□□□□□□□□□□MQClientInstance□□□□□□□□□□□□□□□□□□□□□□□□□□InstanceName□□□□□□□□MQClientInstance□□□

11.3.2 MQClientInstance□□□□

□□□□□□MQClientInstance□□Start□□□□Start□□□□□□□□□□□MQClientInstance□□□□□□□□□11-15□□□

□□□□11-15 MQClientInstance□Start□□

```
public void start() throws MQClientException {
    synchronized (this) {
        switch (this.serviceState) {
            case CREATE_JUST:
                this.serviceState = ServiceState.START_FAILED;
                // If not specified,looking address from name server
                if (null == this.clientConfig.getNamesrvAddr()) {
                    this.mQClientAPIImpl.fetchNameServerAddr();
                }
                // Start request-response channel
                this.mQClientAPIImpl.start();
                // Start various schedule tasks
                this.startScheduledTask();
                // Start pull service
                this.pullMessageService.start();
                // Start rebalance service
                this.rebalanceService.start();
                // Start push service
                this.defaultMQProducer.getDefaultMQProducerImpl().start (false);
                log.info("the client factory [{}] start OK", this.clientId);
                this.serviceState = ServiceState.RUNNING;
                break;
            case RUNNING:
                break;
            case SHUTDOWN_ALREADY:
                break;
            case START_FAILED:
                throw new MQClientException("The Factory object[" +
this.getClientId() + "] has been created before, and failed.", null);
            default:
                break;
        }
    }
}
```

Start□□□□MQClientAPIImpl□□□□□□□□□□□□□□□□□pullMessageService□rebalanceService□□□□□□□□□□□topicRouteTable□brokerAddrTable□□□□□NameServer□□□□□□□□□□□□□□□□ScheduledTask□□□□□□□□MQClientInstance□□□□□□□□□□□□11-16□□□

□□□□11-16 MQClientInstance□□□□□□□□

```
private void startScheduledTask() {
    if (null == this.clientConfig.getNamesrvAddr()) {
        this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
            @Override
            public void run() {
                try {
                    MQClientInstance.this.mQClientAPIImpl
                        .fetchNameServerAddr();
                } catch (Exception e) {
                    log.error("ScheduledTask fetchNameServerAddr " +
                        "exception", e);
                }
            }
        }, 1000 * 10, 1000 * 60 * 2, TimeUnit.MILLISECONDS);
    }
    this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
        @Override
        public void run() {
            try {
                MQClientInstance.this.updateTopicRouteInfoFromNameServer();
            } catch (Exception e) {
                log.error("ScheduledTask " +
                    "updateTopicRouteInfoFromNameServer exception", e);
            }
        }
    }, 10, this.clientConfig.getPollNameServerInterval(), TimeUnit
        .MILLISECONDS);
    this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
        @Override
        public void run() {
            try {
                MQClientInstance.this.cleanOfflineBroker();
                MQClientInstance.this.sendHeartbeatToAllBrokerWithLock();
            } catch (Exception e) {
                log.error("ScheduledTask sendHeartbeatToAllBroker " +
                    "exception", e);
            }
        }
    }, 1000, this.clientConfig.getHeartbeatBrokerInterval(), TimeUnit
        .MILLISECONDS);

    this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
        @Override
        public void run() {
            try {
                MQClientInstance.this.persistAllConsumerOffset();
            } catch (Exception e) {
                log.error("ScheduledTask persistAllConsumerOffset " +
                    "exception", e);
            }
        }
    }, 1000 * 10, this.clientConfig.getPersistConsumerOffsetInterval(),
        TimeUnit.MILLISECONDS);
    this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
        @Override
        public void run() {
            try {
                MQClientInstance.this.adjustThreadPool();
            } catch (Exception e) {
                log.error("ScheduledTask adjustThreadPool exception", e);
            }
        }
    }, 1, 1, TimeUnit.MINUTES);
}
```

[illegible]MQClientInstance[illegible]NameServer[illegible]TopicRoute[illegible]Broker[illegible]Offset[illegible]

11.4 [illegible]

[illegible]Client[illegible]RocketMQ[illegible]DefaultMQPushConsumerImpl[illegible]Consumer[illegible]Class[illegible]MQClientInstance[illegible]RocketMQ[illegible]

□12□　□□□□□□

RocketMQ□Broker□□Master□Slave□□□□□□□□□□□□□□Master□□□□□□□□□□□□□□□□□□□Slave□□□□□□□□Master□□□Slave□□□□□□□□□□□□□□□Master□Slave□□□□□□□□□□□□□□□

12.1 □□□□□□

Slave□□□Master□□□□□□□□□□□□□□□□□□□□□□□□□□TopicConfig□□□ConsumerOffset□□□DelayOffset□SubscriptionGroupConfig□□□Broker□□□□□□□□□□□□□□□□□Slave□□□□□□□□□□□□□□□□□□□12-1□□□

□□□□12-1 Slave□□□□□□□□□□□

```
if (BrokerRole.SLAVE == this.messageStoreConfig.getBrokerRole()) {
    if (this.messageStoreConfig.getHaMasterAddress() != null &&
this.messageStoreConfig.getHaMasterAddress().length() >= 6) {

this.messageStore.updateHaMasterAddress(this.messageStoreConfig.getHaMasterAddres
s());
        this.updateMasterHAServerAddrPeriodically = false;
    } else {
        this.updateMasterHAServerAddrPeriodically = true;
    }
    this.scheduledExecutorService.scheduleAtFixedRate(new Runnable() {
        @Override
        public void run() {
            try {
                BrokerController.this.slaveSynchronize.syncAll();
            } catch (Throwable e) {
                log.error("ScheduledTask syncAll slave exception", e);
            }
        }
    }, 1000 * 10, 1000 * 60, TimeUnit.MILLISECONDS);
}
```

□syncAll□□□□□□syncTopicConfig□□□syncConsumerOffset□□□syncDelayOffset□□□syncSubscriptionGroupConfig□□□□□□□□□□□□□syncConsumerOffset□□□□□□□□□□□□□□□□□□□12-2□□□

□□□□12-2 syncConsumerOffset□□□□

```
public ConsumerOffsetSerializeWrapper getAllConsumerOffset(
    final String addr) throws InterruptedException, RemotingTimeoutException,
    RemotingSendRequestException, RemotingConnectException, MQBroker-Exception {
    RemotingCommand request =
RemotingCommand.createRequestCommand(RequestCode.GET_ALL_CONSUMER_OFFSET, null);
    RemotingCommand response = this.remotingClient.invokeSync(addr, request,
3000);
    assert response != null;
    switch (response.getCode()) {
        case ResponseCode.SUCCESS: {
            return ConsumerOffsetSerializeWrapper.decode(response.getBody(),
ConsumerOffsetSerializeWrapper.class);
        }
        default:
            break;
    }
    throw new MQBrokerException(response.getCode(), response.getRemark());
}
```

---

sysConsumer Offset[illegible]RemotingCommand[illegible]Netty[illegible]Master[illegible]Broker[illegible]Offset[illegible]

12.2 [illegible]

[illegible]Master[illegible]Slave[illegible]CommitLog[illegible]CommitLog[illegible]CommitLog[illegible]Master[illegible]Slave[illegible]Offset[illegible]Consumer[illegible]CommitLog[illegible]Master[illegible]Master[illegible]Slave[illegible]

[illegible]Broker[illegible]org.apache.rocketmq.store.ha[illegible]HAService[illegible]HAConnection[illegible]WaitNotifyObject[illegible]

HAService[illegible]commitLog[illegible]Master[illegible]Slave[illegible]Master[illegible]12-3[illegible]

[illegible]12-3　[illegible]Broker[illegible]HaMasterAddress

```
if (BrokerRole.SLAVE == this.messageStoreConfig.getBrokerRole()) {
    if (this.messageStoreConfig.getHaMasterAddress() != null &&
this.messageStoreConfig
        .getHaMasterAddress().length() >= 6) {

this.messageStore.updateHaMasterAddress(this.messageStoreConfig.getHaMasterAddres
s());
        this.updateMasterHAServerAddrPeriodically = false;
    } else {
        this.updateMasterHAServerAddrPeriodically = true;
    }
```

[illegible]Broker[illegible]Slave[illegible]MasterAddr[illegible]HAService[illegible]HAClient[illegible]connectMaster[illegible]12-4[illegible]

[illegible]12-4　Slave[illegible]Master

```
private boolean connectMaster() throws ClosedChannelException {
    if (null == socketChannel) {
        String addr = this.masterAddress.get();
        if (addr != null) {
            SocketAddress socketAddress =
RemotingUtil.string2SocketAddress(addr);
            if (socketAddress != null) {
                this.socketChannel = RemotingUtil.connect(socketAddress);
                if (this.socketChannel != null) {
                    this.socketChannel.register(this.selector,
SelectionKey.OP_READ);
                }
            }
        }
        this.currentReportedOffset =
HAService.this.defaultMessageStore.getMaxPhyOffset();

        this.lastWriteTimestamp = System.currentTimeMillis();
    }
    return this.socketChannel != null;
}
```

□□□□□□□□□□HAClient□□□□Java NIO□□□□□Master□□□Broker□Master□□□□□□□□□□□□□□□□□12-5□□□

□□□□12-5 □□Slave□HA□□

```
/**
 * Starts listening to slave connections.
 *
 * @throws Exception If fails.
 */
public void beginAccept() throws Exception {
    this.serverSocketChannel = ServerSocketChannel.open();
    this.selector = RemotingUtil.openSelector();
    this.serverSocketChannel.socket().setReuseAddress(true);
    this.serverSocketChannel.socket().bind(this.socketAddressListen);
    this.serverSocketChannel.configureBlocking(false);
    this.serverSocketChannel.register(this.selector, SelectionKey.OP_ACCEPT);
}
```

CommitLog□□□□□□□□□netty command□□□□□□□□□□TCP□□□□□□□□□□□□□□□□□□□□□□Master□Slave□Offset□□□□□□□□□

12.3 sync_master□async_master

sync_master□async_master□□□Broker□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□sync_master□□□□□□□□□Master□□ Broker□□□□□□□□□□□□async_master□□□□□□□□□Master□□ Broker□□□□□□□□□□□□□□□□□Slave□□□□□□□□□□□□□□□□□□ sync_master□□□□□□□□□□□12-6□□□

□□□□12-6 sync_master□□□□□□

```
public void handleHA(AppendMessageResult result,
    PutMessageResult putMessageResult, MessageExt messageExt) {
    if (BrokerRole.SYNC_MASTER == this.defaultMessageStore
        .getMessageStoreConfig().getBrokerRole()) {
        HAService service = this.defaultMessageStore.getHaService();
        if (messageExt.isWaitStoreMsgOK()) {
            // Determine whether to wait
            if (service.isSlaveOK(result.getWroteOffset() + result
                .getWroteBytes())) {
                GroupCommitRequest request = new GroupCommitRequest
                    (result.getWroteOffset() + result
                    .getWroteBytes());
                service.putRequest(request);
                service.getWaitNotifyObject().wakeupAll();
                boolean flushOK =
                    request.waitForFlush(this.defaultMessageStore
                        .getMessageStoreConfig().getSyncFlushTimeout());
                if (!flushOK) {
                    log.error("do sync transfer other node, wait return, " +
                        "but failed, topic: " + messageExt
                        .getTopic() + " tags: "
                        + messageExt.getTags() + " client address: " +
                        messageExt.getBornHostNameString());
                    putMessageResult.setPutMessageStatus(PutMessageStatus
                        .FLUSH_SLAVE_TIMEOUT);
                }
            }
            // Slave problem
            else {
                // Tell the producer, slave not available
                putMessageResult.setPutMessageStatus(PutMessageStatus
                    .SLAVE_NOT_AVAILABLE);
            }
        }
    }
}
```

□CommitLog□□putMessage□□□□□□□handleHA□□□□□□□□□ □□wakeupAll□waitForFlush□□□□□□□□Master□□□□□□□□□□□□□□

Slave[illegible]12-7[illegible]putMessage[illegible]

[illegible]12-7   putMessage[illegible]handleHA

```
public PutMessageResult putMessage(final MessageExtBrokerInner msg) {
    // Set the storage time
    msg.setStoreTimestamp(System.currentTimeMillis());
    // Set the message body BODY CRC (consider the most appropriate setting
    // on the client)
    msg.setBodyCRC(UtilAll.crc32(msg.getBody()));
    // Back to Results
    AppendMessageResult result = null;

    StoreStatsService storeStatsService = this.defaultMessageStore
        .getStoreStatsService();

    String topic = msg.getTopic();
    int queueId = msg.getQueueId();

  ……

    handleDiskFlush(result, putMessageResult, msg);
    handleHA(result, putMessageResult, msg);

    return putMessageResult;
}
```

12.4 □□□□

□□□□□Master□Slave□□□Broker□□Netty□command□□□□□□□□□□□□commitLog□□□□□□□□□□□□□Java NIO□□□□□□□□□□□RocketMQ□□□□□□□□□□□□

[illegible]13[illegible] [illegible]Netty[illegible]

[illegible]RocketMQ[illegible]
TCP[illegible]Socket[illegible]“[illegible]”[illegible]
[illegible]RocketMQ[illegible]Netty[illegible]

13.1 Netty[illegible]

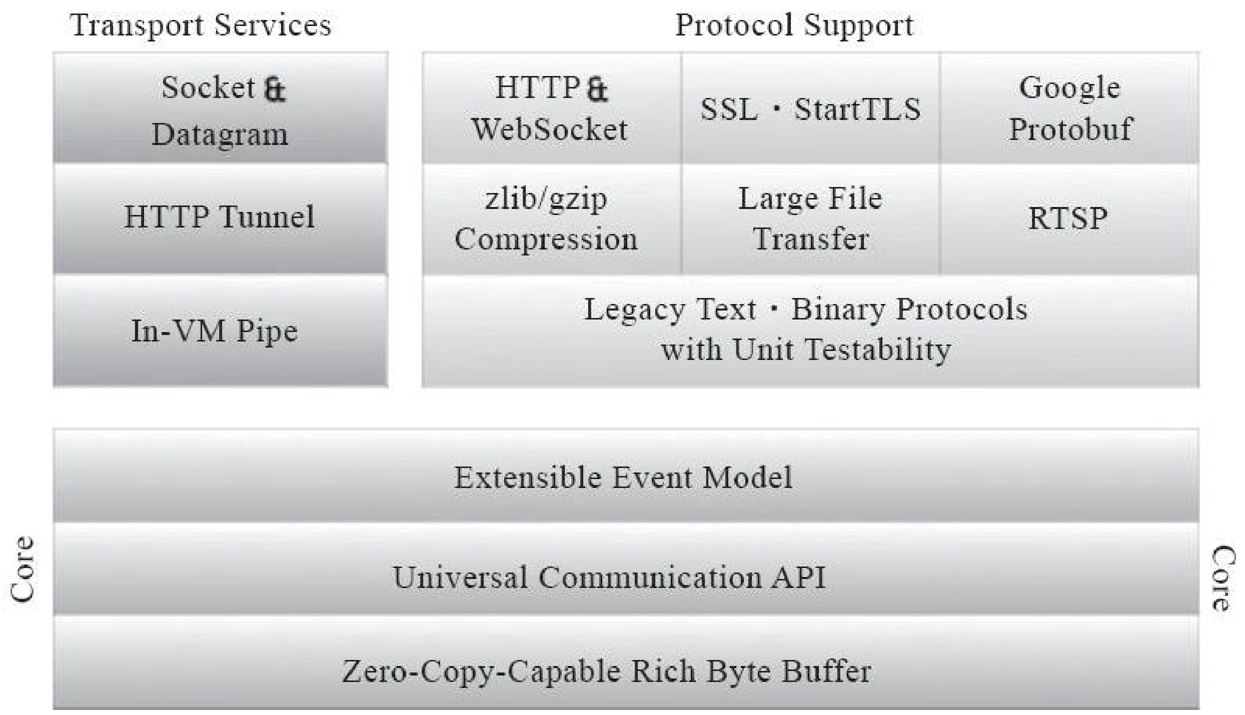
Netty[illegible]Java[illegible]Netty[illegible]/[illegible]

Netty[illegible]FTP[illegible]SMTP[illegible]HTTP[illegible]Netty[illegible]Java[illegible]+[illegible]

[illegible]Netty[illegible]Java NIO[illegible]Channel[illegible]ByteBuffer[illegible]Selector[illegible]Java[illegible]Java NIO[illegible]Client/Server[illegible]Netty[illegible]Netty[illegible]

13.2 Netty□□□□

□□13-1□□□Netty□□□□□□□□□□□□□□□□□□□□□□□□□□□□□JVM□□□Netty□□□□



□13-1 Netty□□□□

13.2.1 [illegible]ByteBuffer

[illegible]CPU[illegible]CPU[illegible]

Netty[illegible]buffer API[illegible]NIO[illegible]ByteBuffer[illegible]buffer[illegible]ByteBuf[illegible]ByteBuffer[illegible]ByteBuf[illegible]ByteBuffer[illegible]

[illegible]ByteBuf[illegible]ByteBuf[illegible]ByteBuf[illegible]read[illegible]skip[illegible]discardReadBytes[illegible]ByteBuffer[illegible]ByteBuf[illegible]

13.2.2　□□□□□I/O□□

□□□Java I/O API□□□□□□□□□□□□□□□□□□□□□□□□□□
java.net.Socket□java.net.DatagramSocket□□□□□□□□□□□□□□□
□□□□□□□□□□□□□Socket□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□Java I/O API□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□TCP/IP□UDP/IP□SCTP□
□□□□□

□□□□□□□□□□Java□□I/O□NIO□API□□□□□□□I/O□OIO□API□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□API□□□□□□□□□
□□□□□□□□□□□□□OIO□Old I/O□API□□□□NIO□□□□OIO□□□□□□□□□□
□□NIO□□□□□□
NIO Selector□□□□□Old I/O□□□□□□□□□□□□□□

Netty□□□□□□Channel□□□□□I/O□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□Netty□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Netty□□□□□□□□□□□□□□□□□□□□□

·□□NIO□TCP/IP□□□io.netty.channel.nio□□

·□□OIO□TCP/IP□□□io.netty.channel.oio□□

·□□OIO□UDP/IP□□□io.netty.channel.oio□□

·□□□□□io.netty.channel.local□□

□□□□□□□□□□□□□□□□□□□□□□□□□□API□□□□□□□□□□□□□□□□□□
□□□□□

13.2.3 [illegible]

[illegible]Netty[illegible] I/O[illegible]Netty[illegible]Netty[illegible]NIO[illegible]

[illegible]Netty[illegible]ChannelPipeline[illegible]ChannelEvent[illegible] ChannelHandler[illegible]Intercepting Filter[illegible]

13.2.4 [illegible]

Netty[illegible]Codec[illegible]SSL/TLS[illegible]HTTP[illegible]

[illegible]Codec[illegible]Netty[illegible]codec[illegible]codec[illegible]codec[illegible]

Netty[illegible]SSL/TLS[illegible]I/O[illegible]NIO[illegible]SSL[illegible]javax.net.ssl.SSLEngine[illegible]SSLEngine[illegible]SSLEngine[illegible]SSLEngine[illegible]Netty[illegible]SslHandler[illegible]SSLEngine[illegible]SslHandler[illegible]ChannelPipeline[illegible]Netty[illegible]StartTlS[illegible]

HTTP[illegible]HTTP[illegible]Netty[illegible]HTTP[illegible]HTTP[illegible]Netty[illegible]HTTP[illegible]HTTP Codec[illegible]HTTP[illegible]HTTP[illegible]web[illegible]

Netty[illegible]WebSockets[illegible]WebSockets[illegible]TCP socket[illegible]Web[illegible]Web[illegible]WebSocket[illegible]IETF[illegible]RFC 6455[illegible]Netty[illegible]RFC 6455[illegible]

[illegible]Netty[illegible]Google Protocol Buffer[illegible]Google Protocol Buffers[illegible]ProtobufEncoder[illegible]ProtobufDecoder[illegible]Google Protocol Buffers[illegible]protoc[illegible]Netty[illegible]codec[illegible]

13.3 Netty[illegible]

13.3.1 Discard[illegible]

[illegible]“Hello[illegible]World[illegible]”[illegible]DISCARD[illegible]DISCARD[illegible]Handler[illegible]Handler[illegible]Netty[illegible]I/O[illegible]13-1[illegible]

[illegible]13-1 DiscardServerHandler[illegible]

```
import io.netty.buffer.ByteBuf;
import io.netty.channel.ChannelHandlerContext;
import io.netty.channel.ChannelInboundHandlerAdapter;
/**
 * [illegible] channel.
 */
public class DiscardServerHandler extends ChannelInboundHandlerAdapter { // (1)
    @Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) { // (2)
        // [illegible]
        ((ByteBuf) msg).release(); // (3)
    }
    @Override
    public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) { //
(4)
        // [illegible]
        cause.printStackTrace();
        ctx.close();
    }
}
```

DiscardServerHandler[illegible]ChannelInboundHandlerAdapter[illegible]ChannelInboundHandler[illegible]ChannelInboundHandler[illegible]ChannelInbound-HandlerAdapter[illegible]

[illegible]chanelRead[illegible]ByteBuf[illegible]

[illegible]DISCARD[illegible]ByteBuf[illegible]release[illegible]

□□□□□□□channelRead□□□□□□□□□□□□□13-2□□□

□□□□13-2　channelRead□□

```
@Override
    public void channelRead(ChannelHandlerContext ctx, Object msg) {
        try {
            // Do something with msg
        } finally {
            ReferenceCountUtil.release(msg);
        }
    }
```

□□□Throwable□□□□Netty□□IO□□□□□□□□□□□□□□□□exceptionCaught□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Channel□□□

□□□□□□□□□□DISCARD□□□□□□□□□□□□□□□□□main□□□□□□□□□DiscardServerHandler□□□□□13-3□□□

□□□□13-3　DiscardServer□□

```
import io.netty.bootstrap.ServerBootstrap;
import io.netty.channel.ChannelFuture;
import io.netty.channel.ChannelInitializer;
import io.netty.channel.ChannelOption;
import io.netty.channel.EventLoopGroup;
import io.netty.channel.nio.NioEventLoopGroup;
import io.netty.channel.socket.SocketChannel;
import io.netty.channel.socket.nio.NioServerSocketChannel;
/**
 * □□□□□□□□□
 */
public class DiscardServer {
    private int port;
    public DiscardServer(int port) {
        this.port = port;
    }
    public void run() throws Exception {
        EventLoopGroup bossGroup = new NioEventLoopGroup(); // (1)
        EventLoopGroup workerGroup = new NioEventLoopGroup();
        try {
            ServerBootstrap b = new ServerBootstrap(); // (2)
            b.group(bossGroup, workerGroup)
             .channel(NioServerSocketChannel.class) // (3)
             .childHandler(new ChannelInitializer<SocketChannel>() { // (4)
                 @Override
                 public void initChannel(SocketChannel ch) throws Exception
                 {
                     ch.pipeline().addLast(new DiscardServerHandler());
                 }
             })
             .option(ChannelOption.SO_BACKLOG, 128)          // (5)
             .childOption(ChannelOption.SO_KEEPALIVE, true); // (6)
            // □□□□□□□□□□□□□□
            ChannelFuture f = b.bind(port).sync(); // (7)
            // □□□□□ Socket □□□
            // □□□□□□□□□□□□□□□□□□□□□□□□□□□□
            f.channel().closeFuture().sync();
        } finally {
            workerGroup.shutdownGracefully();
            bossGroup.shutdownGracefully();
        }
    }
    public static void main(String[] args) throws Exception {
        int port;
        if (args.length > 0) {
            port = Integer.parseInt(args[0]);
        } else {
            port = 8080;
        }
        new DiscardServer(port).run();
    }
}
```

NioEventLoopGroup□□□□□I/O□□□□□□□□□□□□Netty□□□□□□□□EventLoopGroup□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2□NioEventLoopGroup□□□□□□□□□□□□“boss”□□□□□□□□□□□□□□□□□□□“worker”□□□□□□□□□□□□□□□□“boss”□□□□□□□□□□□□□□□□“worker”□□□□□□□□□□□□□□□□□□□□□□□□□□□Channel□□□□□□□EventLoopGroup□□□□□□□□□□□□□□□□□□□□□□□

ServerBootstrap□□□□□NIO□□□□□□□□□□□□□□□□□□□□□Channel□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□NioServerSocketChannel□□□□□□□□Channel□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□Channel□ChannelInitializer□□□□□□□□□□□□□□□□□□□□□□□□□□Channel□□□□□□□□□□□□□□□□DiscardServerHandler□□□□□□□Channel□□□□□□ChannelPipeline□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□pipline□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□Channel□□□□□□□□□□TCP/IP□□□□□□□□□□□□□□Socket□□□□□□□tcpNoDelay□keepAlive□□□□□□□□□

ChannelOption[illegible]ChannelConfig[illegible]ChannelOption[illegible]

option[illegible]NioServerSocketChannel[illegible]childOption[illegible]ServerChannel[illegible]NioServerSocketChannel[illegible]

[illegible]8080[illegible]bind[illegible]

13.3.2 □□□□□□□

□□□□□□□□□□Discard□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□telnet□□□□□□□□□□□□□□telnet localhost 8080□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Discard□□

□□□□□□channelRead□□□□□□□□□□□□□□□□□□□□□DiscardServerHandler□□channelRead□□□□□□□□□□□□□□□□□13-4□□□

□□□□13-4　□□□□channelRead

```
@Override
public void channelRead(ChannelHandlerContext ctx, Object msg) {
    ByteBuf in = (ByteBuf) msg;
    try {
        while (in.isReadable()) { // (1)
            System.out.print((char) in.readByte());
            System.out.flush();
        }
    } finally {
        ReferenceCountUtil.release(msg); // (2)
    }
}
```

□□□□□□□□□□□□□□System.out.println□in.toString□io.netty.util.CharsetUtil.US_ASCII□□□

□□□□□□□in.release□□□□□□□□□telnet□□□□□□□□□□□□□□□□□□□□□□□□□

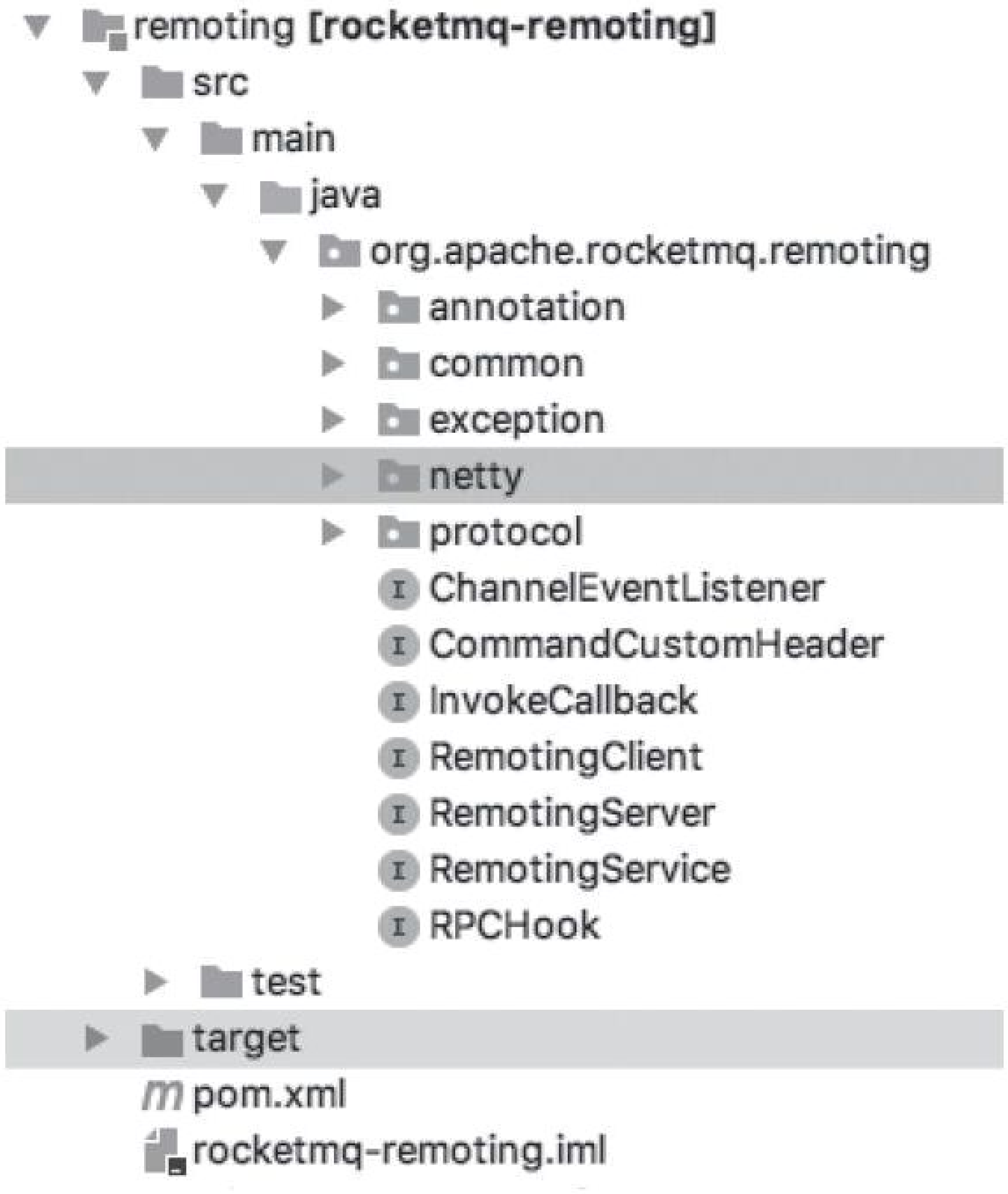
13.4 RocketMQ[illegible]Netty[illegible]

RocketMQ[illegible]Remoting[illegible]Netty[illegible]
RocketMQ[illegible]Netty[illegible]/[illegible]

13.4.1 [illegible]

RocketMQ[illegible]13-2[illegible]

RocketMQ[illegible]RemotingServer[illegible]RemotingClient[illegible]RemotingServer[illegible]13-5[illegible]



```
remoting [rocketmq-remoting]
  src
    main
      java
        org.apache.rocketmq.remoting
          annotation
          common
          exception
          netty
          protocol
          ChannelEventListener
          CommandCustomHeader
          InvokeCallback
          RemotingClient
          RemotingServer
          RemotingService
          RPCHook
    test
  target
  pom.xml
  rocketmq-remoting.iml
```

□13-2　Remoting□□□□□□

□□□□13-5　RemotingService□

```
public interface RemotingServer extends RemotingService {
    void registerProcessor(final int requestCode,
        final NettyRequestProcessor processor,
        final ExecutorService executor);
    void registerDefaultProcessor(final NettyRequestProcessor processor,
        final ExecutorService executor);
    int localListenPort();
    Pair<NettyRequestProcessor, ExecutorService> getProcessorPair(
        final int requestCode);
    RemotingCommand invokeSync(final Channel channel,
        final RemotingCommand request,
        final long timeoutMillis) throws InterruptedException,
        RemotingSendRequestException,
        RemotingTimeoutException;
    void invokeAsync(final Channel channel, final RemotingCommand request,
        final long timeoutMillis,
        final InvokeCallback invokeCallback) throws InterruptedException,
        RemotingTooMuchRequestException, RemotingTimeoutException,
        RemotingSendRequestException;

    void invokeOneway(final Channel channel, final RemotingCommand request,
        final long timeoutMillis)
        throws InterruptedException, RemotingTooMuchRequestException,
        RemotingTimeoutException,
        RemotingSendRequestException;
}
```

RemotingServer□□□□□□□□□localListenPort□registerProcessor□registerDefaultProcessor□registerDefaultProcessor□□□□□□□□□□□□□□□□

RemotingClient□□RemotingServer□□□□□□□□□□□□□updateNameServerAddressList□invokeSync□invokeOneway□updateName-ServerAddressList□□□□□□□NameServer□□□invokeSync□invokeOneway□□□Server□□□□□□□□□□□13-6□□□

□□□□13-6　RemotingClient□

```
public interface RemotingClient extends RemotingService {
    void updateNameServerAddressList(final List<String> addrs);
    List<String> getNameServerAddressList();
    RemotingCommand invokeSync(final String addr, final RemotingCommand request,
        final long timeoutMillis) throws InterruptedException,
        RemotingConnectException,
        RemotingSendRequestException, RemotingTimeoutException;
    void invokeAsync(final String addr, final RemotingCommand request,
        final long timeoutMillis,
        final InvokeCallback invokeCallback) throws InterruptedException,
        RemotingConnectException,
        RemotingTooMuchRequestException, RemotingTimeoutException,
        RemotingSendRequestException;
    void invokeOneway(final String addr, final RemotingCommand request,
        final long timeoutMillis)
        throws InterruptedException, RemotingConnectException,
        RemotingTooMuchRequestException,
        RemotingTimeoutException, RemotingSendRequestException;
    void registerProcessor(final int requestCode,
        final NettyRequestProcessor processor,
        final ExecutorService executor);
    void setCallbackExecutor(final ExecutorService callbackExecutor);
```

```
        boolean isChannelWritable(final String addr);
    }
```

13.4.2 [illegible]

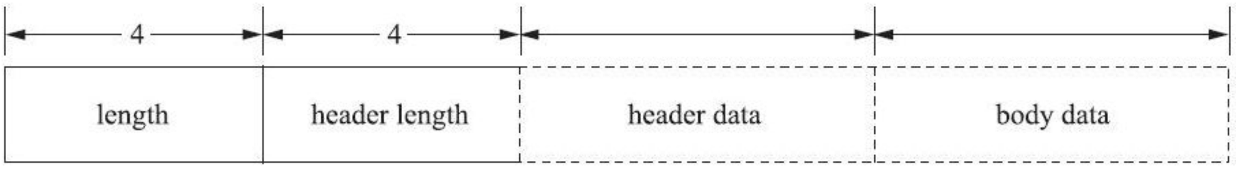
NettyRemotingServer[illegible]NettyRemotingClient[illegible]RemotingServer[illegible]RemotingClient[illegible]invokeSync[illegible]invokeOneway[illegible]NettyRemotingAbstract[illegible]NettyRemotingAbstract[illegible]13-7[illegible]

[illegible]13-7 [illegible]

```
public void processRequestCommand(final ChannelHandlerContext ctx,
    final RemotingCommand cmd) {
    final Pair<NettyRequestProcessor, ExecutorService> matched = this
        .processorTable.get(cmd.getCode());
    final Pair<NettyRequestProcessor, ExecutorService> pair = null ==
        matched ? this.defaultRequestProcessor : matched;
final int opaque = cmd.getOpaque();
------
```

[illegible]processRequestCommand[illegible]Remoting-Command[illegible]

RemotingCommand[illegible]RocketMQ[illegible]13-3[illegible]



[illegible]13-3 RocketMQ[illegible]

[illegible]RocketMQ[illegible]RemotingCommand[illegible]13-8[illegible]

[illegible]13-8 RemotingCommand[illegible]

```
private int code;
private LanguageCode language = LanguageCode.JAVA;
private int version = 0;
private int opaque = requestId.getAndIncrement();
private int flag = 0;
private String remark;
private HashMap<String, String> extFields;
private transient CommandCustomHeader customHeader;
private SerializeType serializeTypeCurrentRPC = serializeTypeConfigInThis-Server;
private transient byte[] body;
```

RocketMQ□□□□□□□□□□□□□□□□□□RemotingCommand□□□□□□□□□□□□□□□□Netty□□□codec□□□□□□□□□□□□□□□□□□□□□□□pipeline□□addLast□new NettyEncoder□□□new NettyDecoder□□…….

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□13-9□□□

□□□□13-9　□□□□□□

```
public RemotingCommand invokeSyncImpl(final Channel channel,
    final RemotingCommand request,
    final long timeoutMillis)
    throws InterruptedException, RemotingSendRequestException,
    RemotingTimeoutException {
    final int opaque = request.getOpaque();
    try {
        final ResponseFuture responseFuture = new ResponseFuture(opaque,
            timeoutMillis, null, null);
        this.responseTable.put(opaque, responseFuture);
        final SocketAddress addr = channel.remoteAddress();
        channel.writeAndFlush(request).addListener(new ChannelFutureListener() {
            @Override
            public void operationComplete(
                ChannelFuture f) throws Exception {
                if (f.isSuccess()) {
                    responseFuture.setSendRequestOK(true);
                    return;
                } else {
                    responseFuture.setSendRequestOK(false);
                }
                responseTable.remove(opaque);
                responseFuture.setCause(f.cause());
                responseFuture.putResponse(null);
                log.warn("send a request command to channel <" + addr +
                    "> failed.");
            }
        });
        RemotingCommand responseCommand = responseFuture.waitResponse
            (timeoutMillis);
        if (null == responseCommand) {
            if (responseFuture.isSendRequestOK()) {
                throw new RemotingTimeoutException(RemotingHelper
                    .parseSocketAddressAddr(addr), timeoutMillis,
                    responseFuture.getCause());
            } else {
                throw new RemotingSendRequestException(RemotingHelper
                    .parseSocketAddressAddr(addr), responseFuture
                    .getCause());
            }
        }
        return responseCommand;
    } finally {
        this.responseTable.remove(opaque);
    }
}
```

□□□RemotingCommand□□□□□□□□□□□□□□□□Channel□□io.netty.channel□Channel□□□□□□□Channel□□Netty□Bootstrap□□□□□□□□□□□□□□□□Channel□□Channel□□□□□□□□□□□□□□□□□□□□□□□□□□□

13.4.3 □□Netty□Server□Client

□□Netty□□□Server□Client□□□□□□□□NettyRemotingServer□NettyRemotingClient□□□□□□□□□ServerBootstrap□□□□□□RocketMQ□□□□□Netty□□Server□□□□□□□□□□13-10□□□

□□□□13-10 ServerBootstrap□□

```
ServerBootstrap childHandler =
    this.serverBootstrap.group(this.eventLoopGroupBoss, this
        .eventLoopGroupSelector)
        .channel(useEpoll() ? EpollServerSocketChannel.class :
            NioServerSocketChannel.class)
        .option(ChannelOption.SO_BACKLOG, 1024)
        .option(ChannelOption.SO_REUSEADDR, true)
        .option(ChannelOption.SO_KEEPALIVE, false)
        .childOption(ChannelOption.TCP_NODELAY, true)
        .childOption(ChannelOption.SO_SNDBUF, nettyServerConfig
            .getServerSocketSndBufSize())
        .childOption(ChannelOption.SO_RCVBUF, nettyServerConfig
            .getServerSocketRcvBufSize())
        .localAddress(new InetSocketAddress(this.nettyServerConfig
            .getListenPort()))
        .childHandler(new ChannelInitializer<SocketChannel>() {
            @Override
            public void initChannel(SocketChannel ch) throws Exception {
                ch.pipeline()
                    .addLast(defaultEventExecutorGroup,
                        HANDSHAKE_HANDLER_NAME,
                        new HandshakeHandler(TlsSystemConfig.tlsMode))
                    .addLast(defaultEventExecutorGroup,
                        new NettyEncoder(),
                        new NettyDecoder(),
                        new IdleStateHandler(0, 0, nettyServerConfig
                            .getServerChannelMaxIdleTimeSeconds()),
                        new NettyConnectManageHandler(),
                        new NettyServerHandler()
                    );
            }
        });
```

ServerBootStrap□BossEventLoop□□□□□□□□NioEventLoopGroup□workerEventLoop□Linux□□□□□□□□3□□□□EpollEventLoopGroup□□□Linux□□□□□□3□□□□NioEventLoopGroup□□□□□□□□□□□□□□□□□NettyEncoder□NettyDecoder□□□Handler□□□Handler□□□□□8□□□DefaultEventExecutorGroup□□

RocketMQ[illegible]Processor[illegible]Executor[illegible]Processor[illegible]Executor[illegible]Broker[illegible]Processor[illegible]13-11[illegible]

[illegible]13-11 [illegible]Processor

```
public void registerProcessor() {
    /**
     * SendMessageProcessor
     */
    SendMessageProcessor sendProcessor = new SendMessageProcessor(this);
    sendProcessor.registerSendMessageHook(sendMessageHookList);
    sendProcessor.registerConsumeMessageHook(consumeMessageHookList);

    this.remotingServer.registerProcessor(RequestCode.SEND_MESSAGE,
        sendProcessor, this.sendMessageExecutor);
    this.remotingServer.registerProcessor(RequestCode.SEND_MESSAGE_V2,
        sendProcessor, this.sendMessageExecutor);
    this.remotingServer.registerProcessor(RequestCode.SEND_BATCH_MESSAGE,
        sendProcessor, this.sendMessageExecutor);
    this.remotingServer.registerProcessor(RequestCode
        .CONSUMER_SEND_MSG_BACK, sendProcessor, this
        .sendMessageExecutor);
```

[illegible]Processor[illegible]org.apache.rocketmq.broker[illegible]BrokerController[illegible]RocketMQ[illegible]

13.5 □□□□

□□□□□RocketMQ□□□□□□□□□□□□□□□□Netty□□□□□□□□□□□□
Netty□□□□□□Netty□□□□□□□□□□□□□□□□□RocketMQ□Netty□□□□
□□
Command□□□□□□□□□□□□□□Processor□Executor□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□